THE UNIVERSITY
of ADELAIDE

Towards Effective and Efficient Semantic Segmentation

Bowen Zhang

A thesis submitted for the degree of
DOCTOR OF PHILOSOPHY
The University of Adelaide

March 5, 2024

Contents

List of Figures

List of Tables

The University of Adelaide

Abstract

Towards Effective and Efficient Semantic Segmentation

by Bowen Zhang

Semantic segmentation, a crucial per-pixel dense prediction task, requires both accuracy and efficiency for effective application in real-world scenarios. This thesis introduces a series of novel methods that address the challenges of semantic segmentation, focusing on achieving high performance while considering computational efficiency.

First, we propose a dynamic neural representational decoder (NRD) to address the heavy computational costs associated with the decoder's upsampling process. Traditional approaches rely on feature pyramid networks or dilated backbones to obtain high-resolution feature maps for semantic segmentation using convolutional network backbones. Instead, we leverage the smoothness prior in the semantic label space and utilize compact neural networks to represent semantic predictions at a patch-level granularity. This significantly reduces computational costs while maintaining competitive performance.

Second, considering the recent advancements in vision transformer networks (ViT), which demonstrate superior performance compared to traditional fully convolution-based networks, this thesis introduces a state-of-the-art decoder framework called Attention-to-Mask (ATM). The ATM leverages attention mechanisms to generate binary masks for semantic segmentation. This decoder not only achieves high performance but also exhibits lightweight characteristics. Additionally, to address the computational cost and redundancy associated with ViT backbones, a Shrunk structure is proposed to enhance efficiency while maintaining the effectiveness of the ATM.

Finally, despite Vision transformers achieving leading performance in various visual tasks, they still suffer from high computational complexity. This issue becomes more pronounced in dense prediction tasks like semantic segmentation, where high-resolution inputs and outputs entail a larger number of tokens involved in computations. While direct token removal has been discussed for image classification tasks, it cannot be extended to semantic segmentation due to the requirement of dense predictions for every patch. To address this challenge, we propose a novel method called Dynamic Token Pruning (DToP) for semantic segmentation, based on the early exit of tokens. Inspired by the human coarse-to-fine segmentation process, we naturally partition the widely adopted auxiliary-loss-based network architecture into multiple stages, with each auxiliary block grading the difficulty level of each token. By leveraging this approach, we can make early predictions for easy tokens, eliminating the need to complete the entire forward pass. Experimental evaluations reveal that the proposed DToP architecture reduces, on average, 20%-35% of the computational cost for current semantic segmentation methods based on plain vision transformers, all while maintaining accuracy.

The effectiveness of the proposed methods is extensively evaluated on benchmark datasets, demonstrating their efficiency and accuracy in achieving state-of-the-art semantic segmentation results.

Declaration of Authorship

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that the copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search, and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Bowen Zhang

June 2023

Acknowledgements

My research journey encountered considerable challenges due to the profound impact of the Covid-19 pandemic. During this challenging period, the unwavering support of my supervisors, classmates, and friends played a crucial role in helping me persevere, and I am sincerely grateful to all of them.

First and foremost, I extend my deep gratitude to Prof. Chunhua Shen, who graciously opened the doors for me to enter the machine learning field as a complete newcomer. Without his support and guidance, my research would never have found its starting point.

Equally deserving of my gratitude is Dr. Zhi Tian, who patiently mentored me from being a newcomer in computer vision to gradually maturing as a researcher and eventually completing my first research work. His expertise and encouragement were instrumental in my growth as a researcher.

I am also equally grateful to Dr. Yifan Liu, who was not only my classmate and co-author but also eventually my supervisor. Her unwavering support and encouragement gave me the strength and courage to persevere during the hardest times of my Ph.D. journey. Her guidance throughout the research process enabled me to navigate through all difficulties and eventually qualify for graduation.

I am also grateful to my co-authors and peers for their valuable contributions and support throughout my research journey. Their collaboration and insights were instrumental in shaping the outcomes of our collective work.

I might also acknowledge the impact of Covid-19, which made pursuing any other career path incredibly challenging during that period. With limited options available, I found myself compelled to focus on research as the most viable and rewarding avenue. Despite the difficulties posed by the pandemic, this unforeseen circumstance ultimately pushed me toward a path of academic pursuit.

Lastly, I extend heartfelt gratitude to my parents, who have unwaveringly supported me throughout this journey. Despite being single, without a defined career, and arriving in my 30s, they have never doubted my pursuit of the dream to become a Doctor. Their belief in me has been a constant source of strength and motivation, inspiring me to persevere through challenges and strive for excellence. Their love and encouragement have been instrumental in shaping my path, and I am deeply grateful for their unwavering faith in my aspirations.

Publications

This thesis contains the following works that have been published or prepared for publication:

- Dynamic Neural Representational Decoders for High-Resolution Semantic Segmentation

  Bowen Zhang, Yifan Liu, Zhi Tian, Chunhua Shen

  The Conference on Neural Information Processing Systems (NeurIPS) 2021

- SegViT: Semantic Segmentation with Plain Vision Transformers

  Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, Yifan Liu

  The Conference on Neural Information Processing Systems (NeurIPS) 2022

- SegViTv2: Exploring Efficient and Continual Semantic Segmentation with Plain Vision Transformers

  Bowen Zhang, Liyang Liu, Minh Hieu Phan, Zhi Tian, Chunhua Shen, Yifan Liu

  under review to International Journal of Computer Vision (IJCV)

- Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation

  Quan Tang, Bowen Zhang, Jiajun Liu, Fagin Liu, Yifan Liu

  International Conference on Computer Vision (ICCV) 2023

In addition, I have the following papers not included in this thesis

- Instance and Panoptic Segmentation Using Conditional Convolutions

  Zhi Tian, Bowen Zhang, Hao Chen, Chunhua Shen

  IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)

- ZegCLIP: Towards Adapting CLIP for Zero-shot Semantic Segmentation

  Ziqin Zhou, Bowen Zhang, Yinjie Lei, Lingqiao Liu, Yifan Liu

  The Conference on Computer Vision and Pattern Recognition (CVPR) 2023

Chapter 1

Introduction

Semantic segmentation is a critical computer vision task that focuses on predicting object categories for each pixel in an image. While the concept behind this task is straightforward, its dense prediction nature poses a challenge in achieving both accurate predictions and computational efficiency.

The pioneering work of Fully Convolutional Networks (FCN) [Long et al., 2015a] introduced a widely adopted framework for semantic segmentation. This framework revolutionized the field by combining a deep convolutional neural network (CNN) as the encoder with a segmentation-oriented decoder, enabling the production of dense predictions. One key advantage of this architecture is the utilization of pre-trained classification models as encoders. These models, trained on large-scale datasets for image classification tasks, possess powerful feature extraction capabilities. By incorporating these pre-trained encoders into semantic segmentation methods, the encoder component effectively performs the bulk of semantic interpretation. Consequently, the segmentation-oriented decoder requires fewer computational resources, leading to improved computational efficiency. The effectiveness of semantic segmentation methods is largely dependent on the ability to adapt the encoder structure in a robust and efficient manner. By fine-tuning the encoder, we can achieve improved alignment with the specific requirements of semantic segmentation, thereby leveraging the valuable knowledge and representations acquired from pre-trained classification models. This adaptation process plays a pivotal role in attaining precise and meaningful predictions in various semantic segmentation tasks.

The primary objective of this thesis is to concentrate on the development of decoder structures that can be seamlessly integrated with different types of backbones for the semantic segmentation task. Additionally, we aim to explore the subsequent tasks that arise as a result of the semantic segmentation process. By addressing the design of the decoder structures, we seek to enhance the overall performance and efficiency of semantic segmentation models across diverse applications and datasets.

1.1 Semantic segmentation decoder for pyramid backbone

Classic classification models like ResNet [He et al., 2016b] have a pyramid structure, where the feature map resolution decreases as the model goes deeper to achieve a larger receptive field, essential for classification. However, this downsized feature map is not suitable for semantic segmentation, which requires pixel-level prediction. To address this challenge, previous methods [Chen et al., 2017a,b, 2018a; Zhao et al., 2017a] adopt dilation convolution in the encoder to acquire a higher-resolution feature map while maintaining a large receptive field. Alternatively, other methods [Kirillov et al., 2019; Li et al., 2020b; Lin et al., 2017a; Yuan et al., 2020] employ a structure similar to Feature Pyramid Network (FPN) [Lin et al., 2017b], gradually upsampling, stacking, and merging feature maps from multiple levels to obtain a larger and semantically-rich

feature map. However, the dilation convolution approach significantly increases the computational cost, often doubling or tripling it. On the other hand, FPN-related methods generally have weaker performance and complex structures compared to dilation convolution methods. As a result, the necessity arises to create a decoder that efficiently leverages the feature maps produced by pyramid-structured encoders, all while maintaining a balance between performance and computational efficiency. More precisely, there's a requirement for an upsampling algorithm capable of closing the distance between high-level, semantically dense yet small-resolution feature maps, and low-level, semantically lacking but large-resolution feature maps. In Chapter 3, we leverage the expressive power of dynamic neural networks and propose a lightweight decoder head that can perform efficient upsampling while maintaining good performance. Our approach aims to address the challenges associated with feature map fusion and resolution enhancement in semantic segmentation models.

1.2 Semantic segmentation decoder for plain backbone

The emergence of the Transformer structure has revolutionized the field of computer vision, particularly with the introduction of the Vision Transformer (ViT) [Dosovitskiy et al., 2021b]. ViT distinguishes itself from CNN-based encoders in two fundamental ways. Firstly, ViT incorporates global attention at every token, enabling a global receptive field that encompasses every location in the feature map. This global attention mechanism allows ViT to capture long-range dependencies effectively. Secondly, ViT's global attention eliminates the need for downsampling in the feature map, thereby preserving a high resolution throughout the network. In contrast, pyramid-structured encoders heavily rely on downsampling to expand the receptive field. The unique combination of global attention and absence of downsampling has contributed to ViT's widespread popularity and improved performance, establishing it as a "plain" structured model. However, the introduction of new encoder structures like ViT brings forth new challenges. There is a pressing need to develop a decoder specifically tailored to the ViT encoder structure, capable of effectively harnessing its unique global attention and plain structure.

There are existing works that targeted solving this problem in Lin et al. [2022]; Ranftl et al. [2021]; Strudel et al. [2021]; Zheng et al. [2021a]. In Chapter 4, we propose a simple yet highly effective mechanism to directly convert the global attention in transformer layers into per-category binary masks. This innovative approach allows the decoder to efficiently utilize the global attention information from ViT and produce accurate per-category predictions. By leveraging this mechanism, we achieve a decoder that strikes a balance between computational efficiency and performance, enabling effective semantic segmentation with ViT-based models.

1.3 Efficient pruning for semantic segmentation

Dense prediction tasks like semantic segmentation often face efficiency challenges as they require predictions for each input pixel location. This makes real-time inference for semantic segmentation more challenging compared to classification tasks. Consequently, achieving efficient semantic segmentation has been a persistent issue in the field. Although the introduction of ViT backbones has improved the semantic interpolation capabilities, ViT structures tend to have redundancy and require more computational resources compared to pyramid counterparts. In the realm of classification tasks, there exist various methods [Liang et al., 2022b; Rao et al., 2021; Yin

et al., 2022] that employ different strategies to reduce the number of tokens as the network goes deeper. These methods are plausible in classification tasks since only the class token is necessary for classification purposes. However, in the context of dense prediction, it is not feasible to simply discard patch tokens, as semantic segmentation requires per-pixel predictions. Hence, there is still ample room for exploration in developing efficient structures on ViT backbones specifically for semantic segmentation tasks.

In Chapter 5, we introduce a novel pruning routine aimed at significantly reducing the computational cost during inference. Inspired by the human coarse-to-fine segmentation process, we divide the commonly used auxiliary-loss-based network architecture into multiple stages, where each auxiliary block assesses the difficulty level of each token. By doing so, we can make early predictions for easy tokens, avoiding the need to complete the entire forward pass. This approach allows us to reduce the computational cost of existing trained semantic segmentation methods by an impressive 20%-35% without compromising performance.

To summarize, this thesis presents two innovative decoder methods for semantic segmentation, specifically addressing the major concerns in the most commonly used encoders (pyramid and plain). Additionally, we thoroughly explore both structural improvements and pruning techniques to enhance inference efficiency and achieve superior results.

Chapter 2

Literature review

2.1 Semantic Segmentation on pyramid encoders

2.1.1 Encoder-decoder Framework

The encoder-decoder architecture is widely used to solve the semantic segmentation task, and almost all the mainstream semantic segmentation methods can be categorized into this family. Typically, the encoder gradually reduces the resolution of feature maps and extracts semantic features, while the decoder is applied to the output features of the encoder to decode the desired semantic labels and recover the spatial resolution. This thesis focuses on the decoder.

2.1.2 Neural network representations

Recently, many works [Mescheder et al., 2019; Peng et al., 2020; Sitzmann et al., 2020] exploit neural networks to represent 3D shapes, which follow the idea that a 3D shape can be represented with a classification model and the 3D shape can be restored by forwarding the 3D coordinates through the classification network. These methods can be viewed as representing the point cloud data with the neural network's parameters.

2.1.3 Dynamic filter networks

Different from traditional convolutions whose filters are fixed during inference once learned, the filters are dynamically generated by another network (namely, the controller). This idea was proposed by [Jia et al., 2016], which enlarges the capacity of the network and captures more content-dependent information such as contextual information. Recently, CondInst [Tian et al., 2022] makes use of dynamic convolutions to implement the dynamic mask heads, which are used to predict the masks of individual instances. In this thesis, we follow in this vein for a different purpose, which is to dynamically generate the parameters of the networks representing local label masks so as to produce high-resolution semantic segmentation results.

2.1.4 upsampling for semantic segmentation

The most commonly used bilinear upsampling can be viewed as the simplest decoder, which assumes that the semantic label maps are smooth to a large extent and the linear interpolation is sufficient to approximate them. Thus, using bilinear upsampling here is effective when the semantic label maps are simple, but the performance is not satisfactory if the label maps are complicated. DeconvNet [Noh et al., 2015] introduces deconvolutional layers in its decoder to step-by-step recover the resolution of the prediction, which can result in much better performance. UPerNet [Xiao et al., 2018] uses an FPN-like structure to fuse feature maps of different scales, and obtains

high-resolution feature maps. DeepLabv3+ [Chen et al., 2018a] designs an effective decoder module that makes use of both encoder-decoder structure and dilation/atrous convolution, which is still one of the most competitive segmentation methods to date, especially in the trade-off between accuracy and computation complexity. CARAFE [Wang et al., 2019] first upsamples feature maps with parameter-free methods and then applies a learnable content-aware kernel mask to the upsampled feature maps. Thus far, despite achieving some success, we believe that there is much room for improvement in terms of taking full advantage of label space prior and designing highly effective and compact decoders for semantic segmentation. The proposed NRD attempts to narrow this gap.

2.2 Semantic Segmentation on ViT encoders

2.2.1 Transformer for Vision

Attention-based transformer models have emerged as powerful alternatives to standard convolution-based networks in the realm of image classification tasks. The original ViT [Dosovitskiy et al., 2021b] represents a plain, non-hierarchical architecture. However, there have been several advancements in the field of hierarchical transformers, such as PVT [Wang et al., 2021a], Swin Transformer [Liu et al., 2021], TWINS [Chu et al., 2021], SegFormer [Xie et al., 2021], and P2T [Wu et al., 2022b]. These hierarchical transformer models inherit certain design elements from convolution-based networks, including hierarchical structures, pooling, and downsampling with convolutions. Consequently, they can be seamlessly employed as direct replacements for convolutional-based networks and can be coupled with existing decoder heads for tasks such as semantic segmentation.

2.2.2 Decoders for ViT encoders

In dense prediction tasks like semantic segmentation, high-resolution feature maps generated by the backbone play a crucial role. In typical hierarchical transformer models, techniques such as FPN [Lin et al., 2017b] or dilated backbone are employed to generate high-resolution feature maps by merging features from different levels. However, when it comes to a plain, non-hierarchical transformer backbone, the resolution remains the same across all layers. SETR [Zheng et al., 2021a] proposed a straightforward approach to address segmentation tasks by treating transformer outputs from the base model in a sequence-to-sequence perspective. Segmenter [Strudel et al., 2021] combines class embeddings and transformer patch embeddings and applies several self-attention layers on the combined tokens to learn discriminative embeddings. In their approach, the class tokens are used as input to the ViT backbone, resulting in increased computational complexity. In contrast, our SegViT introduces the class tokens as input to the ATM, the Attention-to-Mask module, thereby reducing computational costs while still benefiting from the integration of class tokens.

2.2.3 Continual Learning for semantic segmentation

Continual learning (CL) aims to address the issue of forgetting and maintaining the performance on previously learned classes while continuously learning new classes [Chen and Liu, 2016]. Most CL methods propose regularization techniques for convolution-based networks [Douillard et al., 2020; Kang et al., 2022; Li and Hoiem, 2018; Peng et al., 2021] or expand the network architectures to accommodate new tasks [Yan

et al., 2021], thereby avoiding the need to store and replay old data. In recent years, efforts have also emerged to prevent forgetting in Transformer models. Dytox [Douillard et al., 2022] dynamically learns new task tokens, which are then utilized to make the learned embeddings more relevant to the specific task. Lifelong ViT [Wang et al., 2022a] and contrastive ViT [Wang et al., 2022b] introduce cross-attention mechanisms between tasks through external key vectors, and they slow down the changes to these keys to mitigate forgetting. Despite the use of complex mechanisms to prevent forgetting, these methods still require fine-tuning of the network for new classes, which can result in interference with previously learned knowledge.

In the field of semantic segmentation, recent research has been devoted to addressing the forgetting issue in continual learning. However, in addition to forgetting, continual semantic segmentation (CSS) also encounters the problem of 'background shift.' This refers to the situation where foreground object classes from previous tasks are mistakenly classified as background in the current task [Cermelli et al., 2020]. REMINDER [Phan et al., 2022] tackles forgetting in CSS by utilizing class similarity to identify the classes that are more likely to be forgotten. It then focuses on revising those specific classes to mitigate the forgetting problem. RCIL [Zhang et al., 2022b] introduces a two-branch convolutional network, with one branch frozen and the other trained to prevent forgetting. At the end of each learning step, the trainable branch is merged with the frozen branch, which can introduce model interference. However, it is worth noting that existing CSS and CL techniques typically involve fine-tuning certain parts of the network dedicated to the old tasks. Unfortunately, this fine-tuning process can lead to forgetting as the model diverges from the previously learned solution.

2.3 Pruning for Semantic segmentation

2.3.1 Token Reduction

Given that the computational complexity of vision transformers roughly scales quadratically with the length of input sequences, reducing the number of tokens appears to be a direct method for lowering computation expenses. DynamicViT [Rao et al., 2021] observes that an accurate image classification can be obtained by a subset of most informative tokens and proposes a dynamic token specification framework. EViT [Liang et al., 2022b] demonstrates that not all tokens are attentive in multi-head self-attention and reorganizes them based on the attentiveness score with the $[cls]$ token. A-ViT [Yin et al., 2022] computes a halting score for each token using the original network parameter and reserves computing for only discriminative tokens.

These token reduction approaches are carefully designed for image classification based on the intuition that removing uninformative tokens (*e.g.* backgrounds) yields a minor negative impact on the final recognition. However, things changed in semantic segmentation as we are supposed to make predictions on all image patches. Liang *et al.* [Liang et al., 2022a] develop token clustering/reconstruction layers to decrease the number of tokens at middle layers and increase the number before the final prediction. Lu *et al.* [Lu et al., 2023] introduced an auxiliary PolicyNet before transformer layers to guide the token merging operation in regions with similar content. SparseViT [Chen et al., 2023] introduced a pruning routine on Swin Transformer for dense prediction tasks. Differently, we perform token reduction by finalizing the prediction of easy tokens at intermediate layers and reserving computing only for hard tokens in a dynamic manner.

2.3.2 Efficient semantic segmentation methods

In image classification, DVT [Wang et al., 2021b] determines the patch embedding granularity and generates different token numbers based on varying recognition difficulties at the image level. Easy images can be accurately predicted with a mere number of tokens, and hard ones need a finer representation. Going one step further, we base DToP on the assumption that image patches with varying contents represented by tokens are of dissimilar recognition difficulties in semantic segmentation. We can halt easy tokens and reserve only hard tokens for subsequent computing by making early predictions via auxiliary blocks at intermediate layers. As we directly combine the early predictions for easy tokens to form the final recognition results, DToP yields no information loss during token reduction and thus requires no token reconstruction operation, compared with the method proposed by Liang *et al*. [Liang et al., 2022a].

DToP draws inspiration from the deep layer cascade (LC) [Li et al., 2017] but stands out due to two distinct features. Firstly, unlike LC, which is applied to pyramid convolutional neural networks, DToP is utilized with plain vision transformers. The inherently flexible architecture of vision transformers allows DToP to lower computation costs without the need for altering the network's architecture or its operators, in contrast to LC which necessitates specialized region convolution. Secondly, DToP retains the k highest confidence tokens for each semantic category for further computation, ensuring that even categories that are typically easier do not terminate prematurely. This approach aids in the more effective utilization of contextual information.

Statement of Authorship

Title of Paper	Dynamic Neural Representational Decoders for High-Resolution Semantic Segmentation
Publication Status	✓ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Published in NIPS 2021

Principal Author

Name of Principal Author (Candidate)	Bowen Zhang		
Contribution to the Paper	Proposed the ideas, conducted the experiments and wrote part of the paper		
Overall percentage (%)	70		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	10/8/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Yifan Liu		
Contribution to the Paper	Discussion, write the paper and the revision		
Signature		Date	29/08/2023

Name of Co-Author	Zhi Tian		
Contribution to the Paper	Proposed the ideas, write the paper and the revision		
Signature		Date	17/08/2023

Name of Co-Author	Chunhua Shen		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	5/9/2023

Chapter 3

Dynamic Neural Representational Decoders for High-Resolution Semantic Segmentation

3.1 Introduction

In the field of semantic segmentation, overcoming the issue of diminished spatial resolution caused by downsampling within CNN architectures is commonly tackled through the implementation of upsampling decoders. These solutions vary from straightforward bilinear upsampling to elaborate designs that incorporate multi-level features, alongside the employment of dilation convolutions for preserving feature map resolution with a trade-off in computational load. To contend with this challenge, we introduce our novel method. Moving forward to articulate the consequences and practicalities of these techniques, let's explore an illustrative example involving an 8×8 local patch within a binary semantic label space, denoted by $P \in \{0,1\}^{64}$.

Let us consider an 8×8 local patch on a binary semantic label space, denoted by $P \in \{0,1\}^{64}$. If we do not consider any structural correlations in the patch, there would be 2^{64} possibilities for this local patch. However, it is clear to see, for any natural images, the vast majority of the possibilities never exist in the real label map and only a tiny fraction of them are really possible (see Fig. 3.1). Considering the redundancy in the labels, most existing decoders that do not explicitly take this into account would be sub-optimal. This motivates us to design a much more effective decoder by exploiting the prior.

A simple approach is dimensionality reduction techniques. As shown in [Tian et al., 2019], the authors first apply principal component analysis (PCA) to the label patches and compress them into low-dimension compact vectors. Next, the network is required to predict these low-dimension vectors, which are eventually restored into the semantic labels by inverting the PCA process. Their method achieves some success. However, the simplicity and linearity assumption of PCA also limits its performance.

The semantic label masks for natural images are not random and follow some distributions, as shown in Fig. 3.1. Therefore, a good mask representation/decoder must exploit this prior. For computational efficiency, we also want the decoder to be in a compact form. Thus, we require the prior to be effectively learnable from data. Recently, many works [Mescheder et al., 2019; Peng et al., 2020; Sitzmann et al., 2020] exploit neural networks to represent 3D shapes. The work of [Mitchell, 1997] found that neural networks enjoy the inductive bias of *smooth interpolation between data points*, which means that for two points of the same label, the neural networks tend to assign the same label to the points between them as well. As a result, we can conclude that the above idea of representing 3D shapes with neural networks can

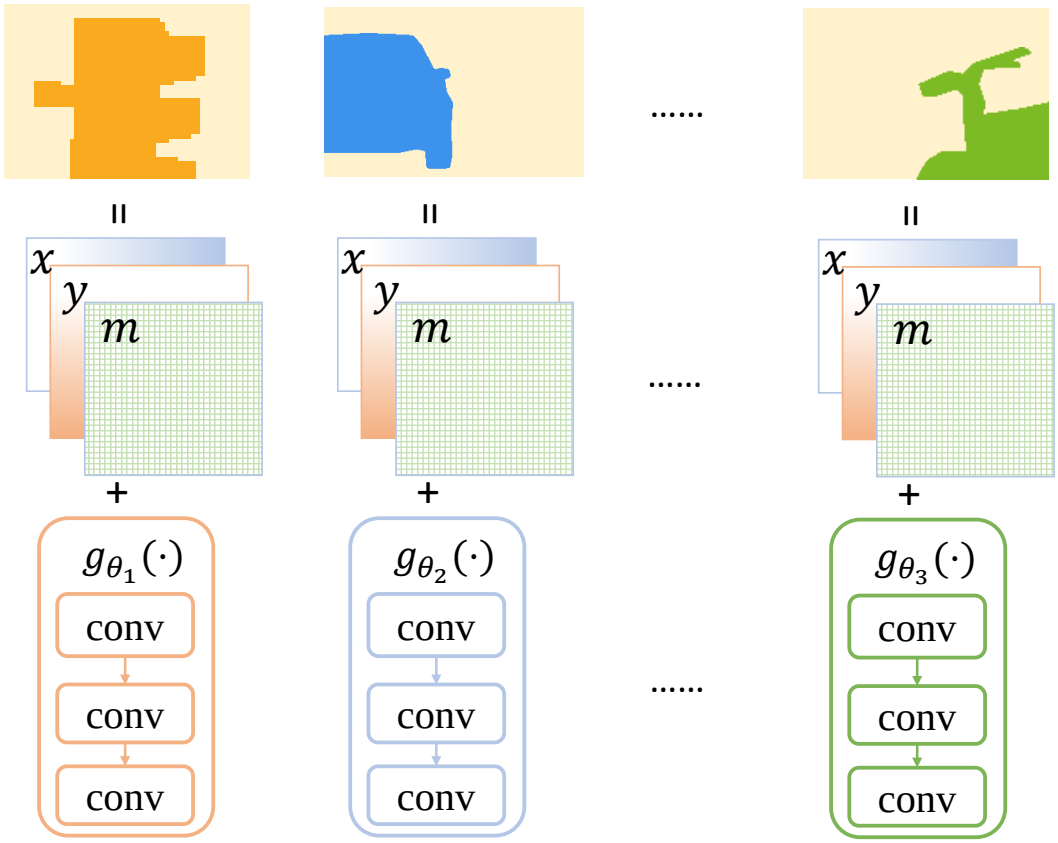


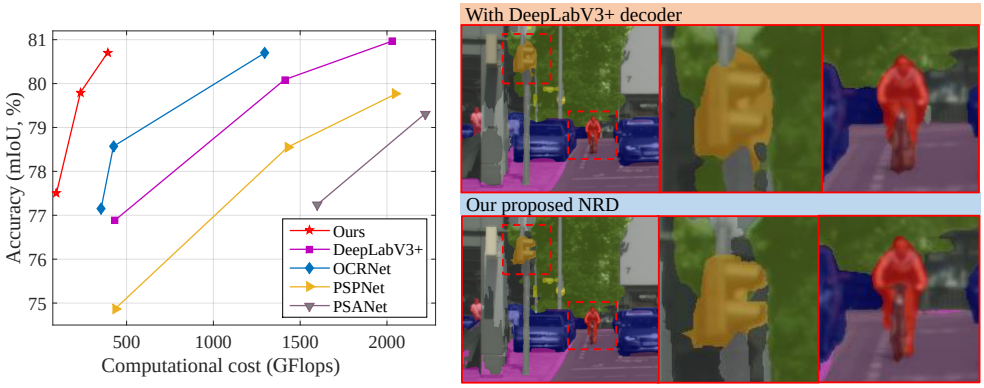
FIGURE 3.1. **The overall concept of our neural representations.** The top row is some examples of the semantic label patches. In the neural representations, each patch is represented with a neural network $g_{\boldsymbol{\theta}}(\cdot)$, as shown in the bottom of this figure. The semantic label patch can be recovered by forwarding the coordinate maps (denoted by x and y in the figure) and the guidance maps (*i.e.*, m in the figure) through the network. As stated in our text, using neural representations for these label patches can implicitly take advantage of the smoothness prior in the semantic label patch.

implicitly leverage the local smoothness prior. Therefore, inspired by these works, we can also represent the local patches of semantic labels with neural networks.

To be specific, as shown in Fig. 3.1, we represent each local label patch by a compact neural network $g_{\boldsymbol{\theta}_i}$ with a few convolution layers interleaved with non-linearities. The semantic labels of a local patch can be obtained by forwarding the corresponding network with (x, y)-coordinate maps and a guidance map m (explained later) as inputs. Furthermore, the parameters $\boldsymbol{\theta}$ of these neural networks, which represent the local label patches, can be dynamically generated with the encoder network in FCNs, and each location on the encoder's output feature maps is responsible for generating the parameters of the neural network representing the specific local label patch surrounding it. The dynamic network makes it possible to incorporate the neural representations into the conventional encoder-decoder architectures and enables a compact design of the decoder, resulting in an end-to-end trainable framework. This avoids the separable learning process as done in [Tian et al., 2019].

Thus, our method is termed *dynamic neural representation decoder* (NRD) for semantic segmentation. We summarize our main contributions as follows.

- We propose a novel decoder that is effective and compact for semantic segmentation, to recover the spatial resolutions. For the first time, we represent the



(A) Accuracy vs. Computational cost (B) Boundary comparison

FIGURE 3.2. (a) Accuracy vs. computational cost on the validation set of Cityscapes. Our proposed NRD can achieve a better trade-off. (b) Comparison between our proposed NRD and the decoder in DeepLabV3+ [Chen et al., 2018a]. We can see that NRD is capable of generating improved boundaries.

local label patches using neural networks and make use of dynamic convolutions to parametrize these neural networks.

- Different from previous methods, which often neglect the redundancy in the semantic label space, our proposed decoder NRD can better take advantage of the redundancy, and thus it is able to achieve *on par* or improved accuracy with significantly reduced computational cost. As shown in Fig. 3.2a, we achieve a better trade-off between computational cost and accuracy compared to previous methods.
- Compared with the decoder used in the classic encoder-decoder model DeeplabV3+ [Chen et al., 2018a], we achieve an improvement of 0.9% mIoU on the Cityscapes dataset with less than 30% computational cost. Moreover, on the trimaps, where only the pixels near the object boundaries are evaluated, a 1.8% improvement can be obtained. This suggests that NRD can substantially improve the quality of the object boundaries.

  Moreover, NRD is even more significant than some methods that use dilated encoders, which usually require 4× more computational cost than ours with similar accuracy. For example, NRD achieves 46.09% mIoU on the competitive ADE20K dataset, which is comparable to that of DeepLabV3+ with a dilated encoder (46.35%) but with only 30% computational cost. We also benchmark our method on the Pascal Context dataset and show excellent performance with much less computational cost.

3.2 Our Method

3.2.1 Overall Architecture

Given an input image $I \in \mathbb{R}^{H \times W \times 3}$, the goal of semantic segmentation is to provide the pixel-level classification score map of the shape of $H \times W \times C$, where C equals the number of categories to be classified into. As mentioned above, mainstream semantic segmentation methods are often based on encoder-decoder architectures. We also

follow this line. Fig. 3.3 shows the overall framework of the proposed model for semantic segmentation.

Our work focuses on the decoder part, and thus we simply make our encoder the same as DeeplabV3+ [Chen et al., 2018a]. The encoder consists of a CNN backbone (*e.g.*, ResNet) and some optional modules such as ASPP [Chen et al., 2017b], which can enhance the output features. By forwarding an input image $I \in \mathbb{R}^{H \times W \times C}$ through the encoder, it generates feature maps with the shape of $H/r \times W/r \times D$, where D is the number of the channels of the feature maps and r is the downsampling ratio of the encoder.

The downsampling ratio is determined by the down-sampling operators in the encoder and can be adjusted by reducing the stride of these down-sampling operators. Dilated convolutions are often used to compensate for the reduction of receptive fields after reducing the strides, with the price of computation overhead. An encoder that reduces the strides and uses dilation convolutions is often referred to as a *dilated encoder*. For example, an encoder based on the standard ResNet backbone produces the feature maps with $r = 32$. Most methods [Chen et al., 2018a; Yuan et al., 2020; Zhang et al., 2018] dilate the encoder and reduce r to 16 or 8. By using the dilated encoder, these methods can output higher-resolution results while the dilated encoder would significantly increase the computational cost. In our work, we do not dilate the encoders (*e.g.*, using $r = 32$) for faster computation and our proposed NRD is expected to better predict the semantic mask at a high resolution.

Let us denote the encoder's output feature maps by $F \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times D}$, whose resolution is $1/32$ of the input image and the desired semantic label map (*i.e.*, the final results). Thus, we make each spatial location on F responsible for a 32×32 local patch surrounding the location and predict the local label map of the patch with our proposed NRD. Finally, the label maps of these patches are merged into the full-resolution segmentation results.

3.2.2 Dynamic Neural Representational Decoders (NRD)

In this section, we provide the details of our NRD and how we generate the parameters for it. The core idea here is to make use of a neural network to represent a local label patch. Thus, given a ground-truth semantic label map $Y \in \{0, 1, ..., C-1\}^{H \times W}$, following the convention, we first convert it to the one-hot label map $Y' \in \{0, 1\}^{H \times W \times C}$, where C is the number of classes. Next, Y' is divided into a number of $H' \times W'$ local patches, and let $P \in \mathbb{R}^{r \times r \times C}$ be one of the patches, where H' and W' are the height and width of the encoder's outputs and r is 32 in our work. Let us take Cityscapes as an example, and thus we have $C = 19$ and $P \in \mathbb{R}^{32 \times 32 \times 19}$. Next, a compact network $g_{\boldsymbol{\theta}}(\cdot)$ is designed to represent the local mask patch P, as shown in Fig. 3.1. To be specific, in our experiment, $g_{\boldsymbol{\theta}}(\cdot)$ is composed of three 1×1 convolutions interleaved with the non-linearity ReLU. Except for the input and output channels, all the hidden layers in $g_{\boldsymbol{\theta}}(\cdot)$ have 16 channels. The output channels of $g_{\boldsymbol{\theta}}(\cdot)$ are equal to the number of classes (*i.e.*, C).

To recover the local patch P, we apply $g_{\boldsymbol{\theta}}(\cdot)$ to a (x, y)-coordinate map $Q = [0 : \frac{1}{s} : 1] \times [0 : \frac{1}{s} : 1] \in \mathbb{R}^{s \times s \times 2}$, where $[0 : \frac{1}{s} : 1]$ is the range from 0 to 1 with step $\frac{1}{s}$ (s is the desired upsampling rate, being 8 in this work) and '$\times$' means the Cartesian multiplication. Since $g_{\boldsymbol{\theta}}(\cdot)$ is composed of 1×1 convolutions, the outputs of $g_{\boldsymbol{\theta}}(\cdot)$ also have size $s \times s$ and can be denoted as $G \in \mathbb{R}^{s \times s \times 19}$. As shown in Fig. 3.3-(b), $g_{\boldsymbol{\theta}}(\cdot)$ takes guidance maps $m \in \mathbb{R}^{s \times s \times C_m}$ as additional inputs. m is generated by applying two convolutional layers to the low-level feature maps, which reduce the channels of the feature maps to C_m, being 16 in this work. We use the same low-level feature

maps as in DeepLabv3+, whose resolutions are 1/4 of the input image. Afterward, a bilinear upsampling is used to upscale G by 4 times to obtain $P' \in \mathbb{R}^{32\times32\times19}$. Next, we compute the loss between P' and P, which, through the back-propagation, adjusts the network's parameter $\boldsymbol{\theta}$ so that P' is as similar to P as possible. In this way, the network parameters $\boldsymbol{\theta}$ can be viewed as the representation of the local semantic label patch P. Although it is possible to remove the bilinear upsampling here and, by increasing the resolution of Q and m, to make the network directly output the desired resolution 32×32, we do not adopt this because using bilinear is sufficient when the upsampling factor is small (*e.g.*, being 4 here). We note that in the above case the network $g_{\boldsymbol{\theta}}(\cdot)$ has 899 parameters in total ($\#weights = (2{+}16){\cdot}16(conv1){+}16{\cdot}16(conv2){+}16{\cdot}19(conv3)$ and $\#biases = 16(conv1) + 16(conv2) + 19(conv3)$).

As shown in previous works [Tian et al., 2019; Wang et al., 2018], each location on the encoder's output feature maps can encode the information of the local patch surrounding it. Therefore, inspired by dynamic filter networks [Yu et al., 2018a], we can use the decoder's output features at each location to dynamically generate the parameters of the representational network for the label patch of the location. To be specific, given the encoder's output feature maps $F \in \mathbb{R}^{H'\times W'\times D}$, where $H' = H/32$, $W' = W/32$ and D are height, width, and the number of channels of F.

Controller. We apply a 3×3 convolution with 512 channels, which is followed by a 1×1 convolution to generate the parameters $\boldsymbol{\theta}$ (shown as the 'controller' in Fig. 3.3). The number of output channels of the convolution is equal to the number of parameters in $\boldsymbol{\theta}$. The generated parameters are then split and reshaped into the weights and biases in $g_{\boldsymbol{\theta}}(\cdot)$, and then $g_{\boldsymbol{\theta}}(\cdot)$ is forwarded to obtain the semantic prediction P'. P' is supervised by the ground-truth label patch P, making the whole framework end-to-end trainable. The overall architecture is shown in Fig. 3.3.

3.3 Experiments

The proposed model is evaluated on three semantic segmentation benchmarks. The performance is measured in terms of intersection-over-union averaged across the present classes (mIoU). We also evaluate the performance near the object boundaries by calculating mIoU on the trimap following [Chen et al., 2018a]. We evaluate our method on the following benchmarks.

ADE20K [Zhou et al., 2017a] is a dataset that contains more than 20K images exhaustively annotated with pixel-level annotation. It has 20,210 images for training and 2,000 images for validation. The number of categories is 150.

PASCAL Context [Mottaghi et al., 2014b] is a dataset with 4,998 images for training and 5,105 images for validation. We use default settings in [MMSegmentation, 2020] that chose the most frequent 59 classes plus one background class (60 classes in total) as the targets.

Cityscapes [Cordts et al., 2016] is a benchmark for semantic urban scene parsing. The training, validation, and test splits contain 2,975, 500, and 1,525 images with fine annotations, respectively. All images from this dataset are 1024×2048 pixels in size.

Implementation details. We use ResNet-50 and ResNet-101 [He et al., 2016b] as our backbone networks and initialize them with the ImageNet pre-trained weights. The training and testing settings as well as data augmentations inherit the default settings in [MMSegmentation, 2020] unless specified. Specifically, for all datasets, we use 'poly' as our learning policy. The initial learning rate is set at 0.01, the weight decay is set to 0.0005 for Cityscapes and ADE20K. For PASCAL Context, the initial

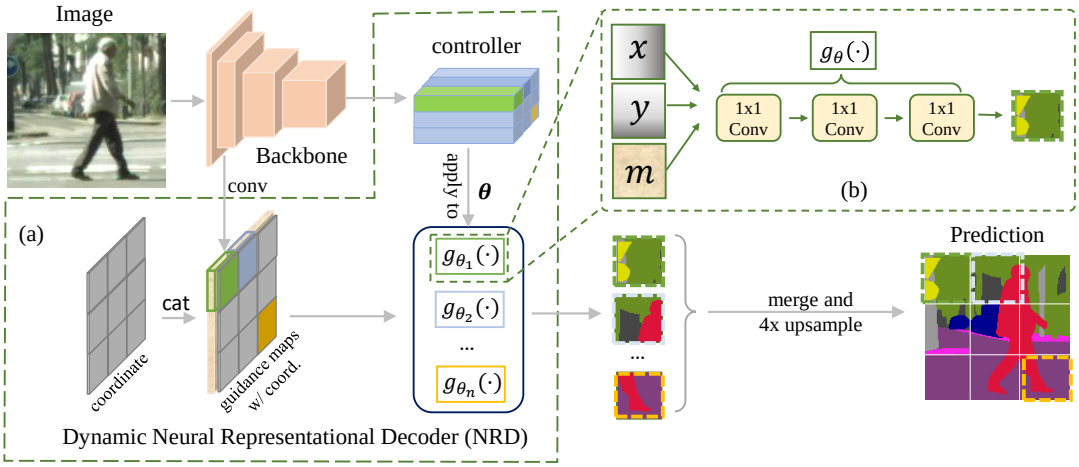


FIGURE 3.3. **The framework of our proposed decoder.** (a) The proposed NRD Module. (b) The details of one of the representational networks $g_{\boldsymbol{\theta}}(\cdot)$. As we can see, we apply the controller to the encoder's output feature maps and generates the parameters $\boldsymbol{\theta}$ of the representational networks. Note that each location on the encoder's output feature maps generates a different set of parameters, which correspond to the representational network of the local patch surrounding the location. Thus we have $H' \times W'$ sets of parameters in total, where H' and W' are the height and width of the encoder's output, respectively. Afterward, the representational networks are fed the (x, y)-coordinate maps and guidance maps m to predict semantic label patches. The guidance maps are generated by applying convolutions to low-level feature maps. We use the same low-level feature maps here as in DeepLabv3+. Finally, these patches are merged into the desired high-resolution segmentation results.

learning rate is 0.004 and the weight decay is 0.0001. We train ADE20K, PASCAL-Context, and Cityscapes for 160k, 80k, and 80k iterations, with the crop size of 512×512, 480×480, and 512×1024, respectively. The training and testing environment is on a workstation with four Volta 100 GPU cards. For test time augmentation, we employ horizontal flip and multi-scale inference. The scale factors are $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$.

3.3.1 Ablation Study

In this section, we conduct the ablation study to show the effectiveness of our proposed NRD. Here, we first compare NRD with the decoder of DeeplabV3+ [Chen et al., 2018a] since it is widely used in practice. Then, we compare it with other decoder methods. Note that when these methods are compared, we use the same encoder for them. Finally, we investigate the hyper-parameters of our model design.

Compared to the DeepLabV3+ decoder. Since we do not use dilation convolutions in our encoder, we also remove the dilation in the DeeplabV3+ encoder for a fair comparison. The results are shown in Table 3.1. As shown in the table, with exactly the same settings, NRD outperforms the decoder in DeeplabV3+ by 0.9% mIoU on the Cityscpaes `val.` split with less than 1/3 computational cost (20.4 vs. 76.4 GFlops), and the total computational cost including the encoder and decoder is reduced from 290.6 to 234.6 GFlops. In addition, on the trimap, NRD is 1.8% mIoU better than the DeeplabV3+ decoder, which suggests that our method is able to produce boundaries of higher quality.

Compared to the bilinear decoder. We also compare our method with the simplest decoder which uses a 1×1 convolution to map the outputs of the encoder

TABLE 3.1. **Our proposed NRD vs. the DeepLabV3+ decoder and bilinear decoder** on the Cityscapes `val.` split. All models use the same encoder and are trained with 84K iterations and 512×1024 crop size. The GFlops is measured with the original image size 1024×2048. All the GFlops in this paper are measured at single-scale inference. $\text{GFlops}^{\text{dec}}$ indicates the GFlops for decoders only.

Method	Backbone	Low-level	$\text{GFlops}^{\text{dec}}$	GFlops	mIoU (%)	Trimap mIoU (%)
Decoder	ResNet-50	stage2	76.4	290.6	78.9	49.8
NRD (**Ours**)	ResNet-50	stage2	20.4	234.6	**79.8** (+0.9)	**51.6** (+1.8)
Bilinear decoder	ResNet-50	None	1.3	215.5	74.7	41.2
NRD (**Ours**)	ResNet-50	None	2.6	216.8	**78.2** (+3.5)	**46.6** (+5.4)

TABLE 3.2. Comparison of different upsampling methods using ResNet50 as backbones on the Cityscapes `val.` split. All methods are trained for $84k$ iterations. The GFlops is measured at single scale inference with a crop size of 1024×2048. The proposed NRD outperforms previous decoders.

Method	GFlops	Params	mIoU (%)
CARAFE	203.0	36.3	72.1
DUC	336.1	110.8	74.7
NRD (Ours)	203.2	36.6	**75.0**

TABLE 3.3. Ablation results on the Cityscapes validation set. C_r is the number of channels of the 1×1 convolutions in $g_{\boldsymbol{\theta}}(\cdot)$. C_m is the number of channels of the guidance map. The accuracy is not very sensitive to these parameters and in general 16 channels for both C_r, C_m lead to marginally better results.

NRD variants						
C_r	8	**16**	32	16	**16**	16
C_m	16	**16**	16	8	**16**	32
mIoU	79.4	**79.8**	79.6	79.5	**79.8**	79.0

to the desired segmentation predictions and then simply uses the bilinear upsampling to upscale the predictions to the desired resolutions. Again, both encoders' output resolutions are $1/32$ of the input image. To make a fair comparison, we also remove the guidance map in NRD (*e.g.*, the low-level features). Thus, only coordinate maps are taken as the input of NRD. As shown in Table. 3.1, NRD surpasses the bilinear decoder by a large margin (+3.5% mIoU). Note that although NRD has a higher computational cost than the bilinear decoder (1.3 vs. 2.6 GFlops), the overall computational cost is almost the same (215.5 vs. 216.8 GFlops) as most of the computational cost is in the encoder. Additionally, the mIoU on the trimap is improved by 5.4%.

Compared to other decoder methods. We also compare NRD with some other decoder methods. ResNet-50 is used as the backbone and we do not use the dilated encoders in all these methods. The results are shown in Table 3.2. As shown in the table, compared to CARAFE [Wang et al., 2019], we improve the mIoU on Cityscapes from 72.1% to 75.0% with similar computational complexity (203.0 vs. 203.2 GFlops) and the number of parameters. In addition, compared to DUC [Wang et al., 2018], which outputs multiple channels and uses the "depth-to-space" operation to increase the spatial resolutions, our NRD is superior to it (75.0% vs. 74.7% mIoU) with only 60% computational complexity (203.2 vs. 336.1 GFlops) and ∼33% parameters.

Ablation study of architectures of NRD. Here, we investigate the hyper-parameters of our NRD. Table. 3.3 shows the performance as we vary the number of channels C_r of the representational network $g_{\boldsymbol{\theta}}(\cdot)$. As we can see in the table, the performance is not very sensitive to the number of channels (within 0.4% mIoU). We also experiment by varying the number of channels C_m of the guidance map m. As shown in Table. 3.3, using $C_m = 16$ can result in slightly better performance than $C_m = 8$ (79.8% vs. 79.5% mIoU), but increasing C_m to 32 cannot improve the

TABLE 3.4. NRD results with various inputs to the representational network $g_{\boldsymbol{\theta}}(\cdot)$. 'Guidance map': use of the guidance map as the inputs to the representational network or not; 'Coord. map': use of the coordinate maps or not. The experimental results are evaluated on the Cityscapes `val.` split. We can see that the guidance map is critical to the segmentation accuracy at the object boundaries (see the trimap mIoU).

Method	Guidance map	Coord. map	mIoU (%)	Trimap mIoU (%)
NRD		✓	78.2	46.6
NRD	✓		78.3	50.5
NRD	✓	✓	**79.8**	**51.5**

performance further.

Table. 3.4 shows the effect of the inputs to the representational network $g_{\boldsymbol{\theta}}(\cdot)$. As we can see, if no guidance maps are given and $g_{\boldsymbol{\theta}}(\cdot)$ only takes as input the coordinate maps, NRD can already achieve decent performance (78.2% mIoU), which is already much better than the bilinear decoder as shown in Table. 3.1. In addition, if $g_{\boldsymbol{\theta}}(\cdot)$ only takes the guidance map as input, NRD can achieve similar performance (78.3% mIoU). However, it can be seen that there is a significant improvement in the trimap mIoU (+3.9% mIoU), which suggests that the guidance map plays an important role in preserving the details. Finally, if both the coordinate maps and the guidance maps are used, NRD can achieve the best performance (79.8% mIoU).

3.3.2 Comparisons with state-of-the-art methods

In this section, we compare our method with other state-of-the-art methods on three datasets: ADE20K, PASCAL-Context, and Cityscapes.

ADE20K. Table 3.9 shows the comparisons with state-of-the-art methods on ADE20K. Our method achieves 45.62% in terms of mIoU with ResNet-101 as the backbone. It is 0.95% better than the recent SFNet [Li et al., 2020b], with the same ResNet-101 backbone. Besides, due to the strong ability of NRD to recover the spatial information, we do not need to use the multi-paths complex decoder as in SFNet, and thus our method only spends 50% computational cost of SFNet. Our method is also better than other methods with dilated encoders, including DMNet [He et al., 2019], ANLNet [Zhu et al., 2019], CCNet [Huang et al., 2019] and EncNet [Zhang et al., 2018], and needs only 20% $\sim$ 30% computational cost of these methods. Additionally, by using a larger backbone ResNext-101, our performance can be further improved to 46.09% mIoU. Note that even with the larger backbone, our method still has much lower computational complexity than other methods with dilated encoders. As a result, we can achieve competitive performance among state-of-the-art methods with significantly less computational cost.

PASCAL-Context. Table 3.6 shows the results on the PASCAL-Context dataset. We follow HRNet [Sun et al., 2019] to evaluate our method and report the results under 59 classes (without background) and 60 classes (with background). Our methods achieve 54.1% (59 classes) and 49.0% (60 classes) mIoU. The results are even better than the sophisticated high-resolution network HRNet with $\sim$50% computational complexity (42.9 vs. 82.7 GFlops). Note that HRNet stacks some hourglass networks and is much more complicated than ours. Our method also achieves better results with less computational cost than other methods, as shown in the table.

Cityscapes. Table 3.7 shows the performance of our method on the Cityscapes `test` split. We train our model with the `trainval` split and only the fine annotations. As we can see, the proposed method can achieve competitive performance with much

TABLE 3.5. Experiment results on the ADE20K `val.` split. The GFlops is measured at single-scale inference with a crop size of 512×512. 'ms' means that mIoU is calculated using multi-scale inference. * means that results are re-implemented by [MMSegmentation, 2020]. Note that compared to the DeepLabv3+, we achieve similar performance (46.09% vs. 46.35% mIoU) with ∼ 30% computational complexity (87.9 vs. 255.1 GFlops). Speed (frames per second, FPS) is measured with the same input size as the single scale inference on an RTX 3090 GPU.

Method	Backbone	Dilated encoder	GFlops	mIoU (%)	mIoU 'ms' (%)	FPS
PSPNet [Zhao et al., 2017b]	ResNet-50	✓	178.8	41.68	42.78	30.01
PSANet [Zhao et al., 2018]	ResNet-50	✓	194.8	41.92	42.97	25.60
EncNet [Zhang et al., 2018]	ResNet-50	✓	>100	-	41.11	33.34
CFNet [Zhang et al., 2019]	ResNet-50	✓	>100	-	42.87	-
RGNet [Yu et al., 2020a]	ResNet-50	✓	>100	-	44.02	-
CPNet [Yu et al., 2020b]	ResNet-50	✓	208.6	43.92	44.46	27.96
DeepLabv3+* [Chen et al., 2018a]	ResNet-50	✓	177.5	43.95	44.93	29.23
PSPNet [Zhao et al., 2017b]	ResNet-101	✓	256.4	41.96	43.29	20.25
PSPNet [Zhao et al., 2017b]	ResNet-269	✓	-	43.81	44.94	-
PSANet [Zhao et al., 2018]	ResNet-101	✓	272.5	42.75	43.77	18.11
EncNet [Zhang et al., 2018]	ResNet-101	✓	>180	-	44.65	21.89
CFNet [Zhang et al., 2019]	ResNet-101	✓	>180	-	44.89	-
CCNet [Huang et al., 2019]	ResNet-101	✓	>180	-	45.22	-
ANLNet [Zhu et al., 2019]	ResNet-101	✓	>180	-	45.24	-
GFFNet [Li et al., 2020c]	ResNet-101	✓	>180	-	45.33	-
DMNet [He et al., 2019]	ResNet-101	✓	>180	-	45.5	-
RGNet [Yu et al., 2020a]	ResNet-101	✓	>180	-	45.8	-
CPNet [Yu et al., 2020b]	ResNet-101	✓	286.3	45.39	46.27	18.25
DeepLabv3+* [Chen et al., 2018a]	ResNet-101	✓	255.1	**45.47**	**46.35**	19.74
SFNet [Li et al., 2020b]	ResNet-50		83.2	-	42.81	27.64
SFNet [Li et al., 2020b]	ResNet-101		102.7	-	44.67	21.95
EfficientFCN [Liu et al., 2020]	ResNet-101		60.5	-	45.28	53.87
OCRNet [Yuan et al., 2020]	HRNetV2-W48		164.8	-	45.66	16.39
NRD (**Ours**)	ResNet-101		**49.0**	44.01	45.62	**54.06**
NRD (**Ours**)	ResNeXt-101		87.9	**44.34**	**46.09**	34.88

less computational complexity. Compared to the recent RGNet [Yu et al., 2020a], our method achieves comparable performance with less than 30% computational cost (>1500 vs. 390.0 GFlops). Our method also has competitive performance with SFNet less than 50% computational complexity (821.2 vs. 390 GFlops). In addition, it is worth noting that our method based on ResNet-50 can have better performance than ResNet-18 based SFNet (79.5% vs. 80.0% mIoU) while having even less computational complexity (234.6 vs. 243.9 GFlops). This suggests that our proposed method has a better speed-accuracy trade-off as shown in Fig. 3.2a.

3.4 More Evaluation and Demo results

3.4.1 Additional Evaluation Results

Evaluation on Cityscapes. 3.8 shows the comparison of mIOU performance for different methods on Cityscapes `val` split. The GFlops and the mIOU performance are all measured in the same structure implemented in [MMSegmentation, 2020]. NRD performs better in terms of accuracy-computation trade-off. NRD usually has the best performance among methods that are of similar computational cost. Even if compared with the methods that use dilated backbones which have 5 times of the computational cost, NRD is still competitive in accuracy.

Evaluation on ADE20K. We further use the recent transformer-based backbone SegFormer Xie et al. [2021] as the encoder to show the ability of the proposed method. The experiments are conducted on the ADE20K dataset. We replace the lightweight

TABLE 3.6. Semantic segmentation results on the PASCAL-Context `val.` split. mIoU_{59}: mIoU averaged over 59 classes (without background). mIoU_{60}: mIoU averaged over 60 classes (59 classes plus background). Both metrics were used in the literature, and we report both for thorough comparisons. Following published methods, we report the results with multi-scale inference (denoted by 'ms'). The GFlops is measured at single scale inference with a crop size of 480×480. 'Dilated-$*$': using dilated encoders.

Method	Backbone	GFlops	mIoU_{59} (ms)	mIoU_{60} (ms)	FPS
FCN-8s [Long et al., 2015a]	VGG-16	-	-	35.1	-
HO-CRF [Arnab et al., 2016]	-	-	-	41.3	-
Piecewise [Lin et al., 2016]	VGG-16	-	-	43.3	-
DeepLab-v2 [Chen et al., 2017a]	Dilated-ResNet-101	-	-	45.7	-
RefineNet [Lin et al., 2017a]	ResNet-152	-	-	47.3	-
UNet++ [Zhou et al., 2018]	ResNet-101	-	47.7	-	-
PSPNet [Zhao et al., 2017b]	Dilated-ResNet-101	157.0	47.8	-	22.45
Ding *et al.* [Ding et al., 2018]	ResNet-101	-	51.6	-	-
EncNet [Zhang et al., 2018]	Dilated-ResNet-101	192.1	52.6	-	24.66
HRNet [Sun et al., 2019]	HRNetV2-W48	82.7	54.0	48.3	19.90
GFFNet [Li et al., 2020c]	Dilated-ResNet-101	-	54.3	-	-
EfficientFCN [Liu et al., 2020]	ResNet-101	52.8	55.3	-	47.8
OCRNet [Yuan et al., 2020]	HRNetV2-W48	143.9	56.2	-	17.11
NRD **(Ours)**	ResNet-101	**42.9**	**54.1**	**49.0**	**49.60**

TABLE 3.7. Experiment results on the Cityscapes `test` split. 'ms' means that mIoU is calculated using multi-scale inference. The GFlops is measured at single scale inference with a crop size of 1024×2048.

Method	Backbone	GFlops	mIoU	mIoU (ms)	FPS
PSPNet [Zhao et al., 2017b]	Dilated-ResNet-101	2049.0	-	78.4	3.36
AAF [Ke et al., 2018]	Dilated-ResNet-101	>1500	-	79.1	-
DFN [Yu et al., 2018a]	Dilated-ResNet-101	>1500	-	79.3	-
PSANet [Zhao et al., 2018]	Dilated-ResNet-101	2218.6	-	80.1	2.86
RGNet [Yu et al., 2020a]	Dilated-ResNet-101	>1500	-	81.5	-
DeepLabV3+ [Chen et al., 2018a]	Dilated-ResNet-101	2032.3	-	81.3	-
DANet [Fu et al., 2019b]	Dilated-ResNet-101	2214.7	-	81.5	-
GFFNet [Li et al., 2020c]	Dilated-ResNet-101	>1500	-	82.3	-
BiSeNet [Yu et al., 2018b]	ResNet-101	>360	-	78.9	-
SFNet [Li et al., 2020b]	ResNet-18	243.9	78.9	79.5	13.6
HRNet [Sun et al., 2019]	HRNetV2-W48	748.7	-	81.6	7.86
SFNet [Li et al., 2020b]	ResNet-101	821.2	-	**81.8**	4.62
NRD (**Ours**)	ResNet-50	234.6	78.9	80.0	18.17
NRD (**Ours**)	ResNet-101	390.0	79.3	80.5	12.23

TABLE 3.8. Accuracy and computational costs of different networks on Cityscapes `val.` split. The model mIoU is measured at single-scale inference. The GFlops is measured at single scale inference with a crop size of 1024 × 2048. Note that comparison results are quoted from MMSegmentation [2020] which provides a re-implementation of all the models in the table.

Method	backbone	GFlops	mIOU (%)
PSANet	Dilated-ResNet-50	1597.15	77.24
PSANet	Dilated-ResNet-101	2218.62	79.31
PSPNet	Dilated-ResNet-18	434.11	74.87
PSPNet	Dilated-ResNet-50	1427.47	78.55
PSPNet	Dilated-ResNet-101	2048.95	79.76
DeepLabv3+	Dilated-ResNet-18	433.9	76.89
DeepLabv3+	Dilated-ResNet-50	1410.86	80.09
DeepLabv3+	Dilated-ResNet-101	2030.3	**80.97**
OCRNet	HRNetV2p-W18-S	353.47	77.16
OCRNet	HRNetV2p-W18	424.29	78.57
OCRNet	HRNetV2p-W48	1296.77	80.7
NRD (**Ours**)	ResNet-18	**95.7**	77.5
NRD (**Ours**)	ResNet-50	234.6	79.8
NRD (**Ours**)	ResNet-101	390.0	80.7

TABLE 3.9. Experiment results on the ADE20K *val.* split. The GFlops is measured at single-scale inference using crop sizes provided in the table. 'ms' means that mIoU is calculated using multi-scale inference.

Method	Backbone	Crop size	GFlops	mIoU (%)	mIoU 'ms' (%)
SegFormer	MiT-B0	512 × 512	8.4	37.4	38.0
NRD	MiT-B0	512 × 512	7.9	**38.1**	**40.1**
SegFormer	MiT-B1	512 × 512	15.9	42.2	43.1
NRD	MiT-B1	512 × 512	14.7	**42.9**	**44.3**
SegFormer	MiT-B2	512 × 512	62.4	46.5	47.5
NRD	MiT-B2	512 × 512	24.5	**46.8**	**48.2**
SegFormer	MiT-B5	640 × 640	183.3	51.0	51.8
NRD	MiT-B5	640 × 640	124.2	**51.2**	**51.9**

all-MLP decoder proposed by Segformer with NRD and follow all training settings in Xie et al. [2021]. The results show that using the exactly same backbone, the performance of NRD is competitive with much lighter computation.

3.4.2 More Visualization Results

Comparison with bilinear upsampling. Figure. 3.4 shows the result comparison between the bilinear decoder and NRD decoder. Note that, the results are generated from feature maps that are $1/32$ of the input scale. Thus, the results can represent the effectiveness comparison between the bilinear method and NRD. From the illustration, we can see that it is inevitable for the decoder to lose some details during a 32 times upsampling. However, NRD clearly preserves more details than the bilinear interpolation method. These two upsampling schemes have similar computation costs.

Comparison with the DeeplabV3+ decoder. Figure. 3.5 shows more comparison between the DeeplabV3+ Chen et al. [2018a] decoder and NRD. The computational cost of the decoder part is 76.4 (DeeplabV3+) vs. 20.4 (NRD) GFlops. From

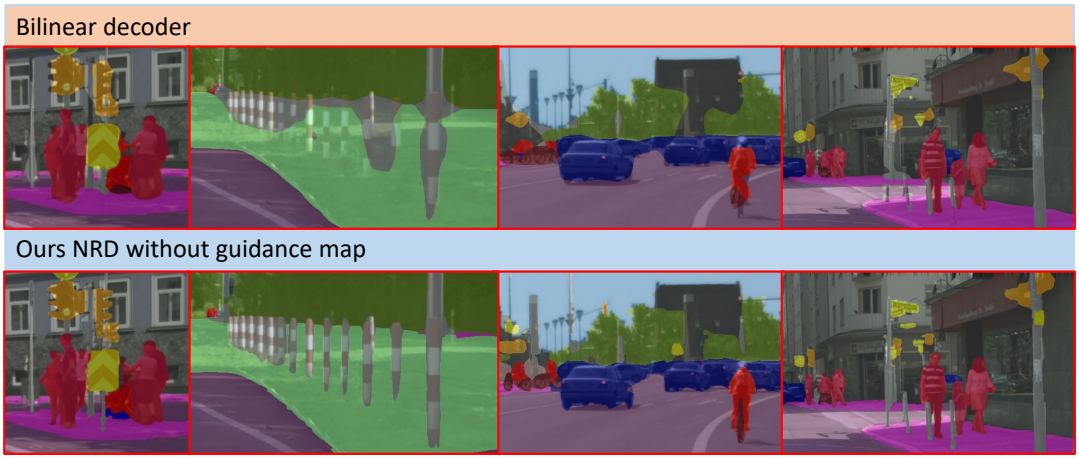


FIGURE 3.4. Comparison between the bilinear decoder and the NRD decoder without guidance map on the Cityscapes dataset. We can see that there is a significant improvement in the boundary region.

the figure, we can see that in various scenes, NRD shows superior segmentation results than DeeplabV3+.

Competitive segmentation results on ADE20K and PASCAL-Context. Figure. 3.6 shows the segmentation results on ADE20K produced by NRD using ResNeXt101 backbone with multi-scale inference. We can see that in various scenes, including the bedroom, the toilet, and some outdoor scenes, NRD can generate satisfactory segmentation results. It can also perform well at boundaries such as the human legs and the poles. Figure. 3.7 is the segmentation results on PASCAL-Context produced by NRD using Resnet101 backbone with multi-scale inference (60 classes).

Detailed illustration of the NRD module. Figure. 3.8 shows how an input image is processed in NRD. For the NRD structure, the guidance maps that are generated from the low-level features and the coordinate maps are concatenated together. These feature maps are served as the input to the NRD structure. We attribute each patch of the feature maps with a representational network $g_{\boldsymbol{\theta}}(\cdot)$ whose parameters are dynamically generated by the controller. In Figure. 3.8 each block surrounded by white lines represents a patch that is processed by a particular $g_{\boldsymbol{\theta}}(\cdot)$. The result is then directly used as the output of the decoder without the use of more convolutions to 'refine' the results.

3.5 Conclusion

We have proposed a compact yet very effective decoder, termed Neural Representational Decoders (NRD), for the semantic segmentation task. For the first time, we use the idea of neural representations for designing the segmentation decoder, which is able to better exploit the structure in the semantic segmentation label space. To implement this idea, we dynamically generate the neural representations with dynamic convolution filter networks so that the neural representations can be incorporated into the standard encoder-decoder segmentation architectures, enabling end-to-end training. We show on a number of semantic segmentation benchmarks that our method is highly efficient and achieves state-of-the-art accuracy. We believe that our method can be a strong decoder in high-resolution semantic segmentation and may inspire other dense prediction tasks such as depth estimation and super-resolution. Last but not least, our method still has some limitations. One of the limitations is that the

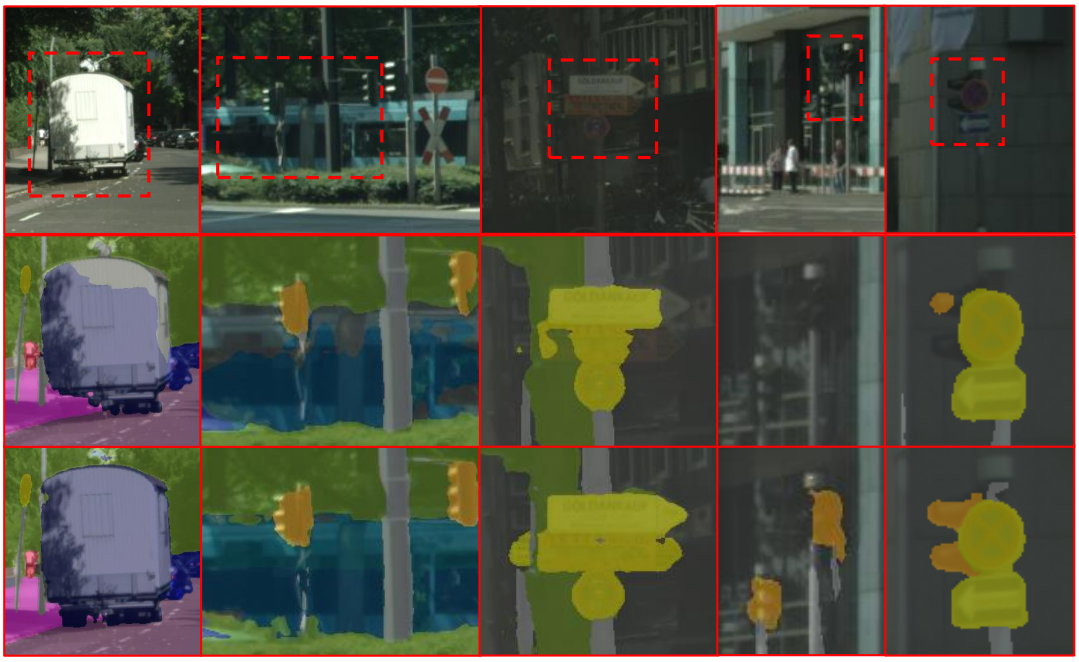


FIGURE 3.5. Visualization results on Cityscapes. From top to bottom: Input image; results of DeeplabV3+; and results of NRD. The single scale GFlops of these two methods are 293.6 (DeeplabV3+) vs. 234.6 (NRD). Our NRD requires less computation, yet the segmentation accuracy is superior.

dynamic filter networks have not been well-supported in some mobile devices, which might restrict the applicability of this method.

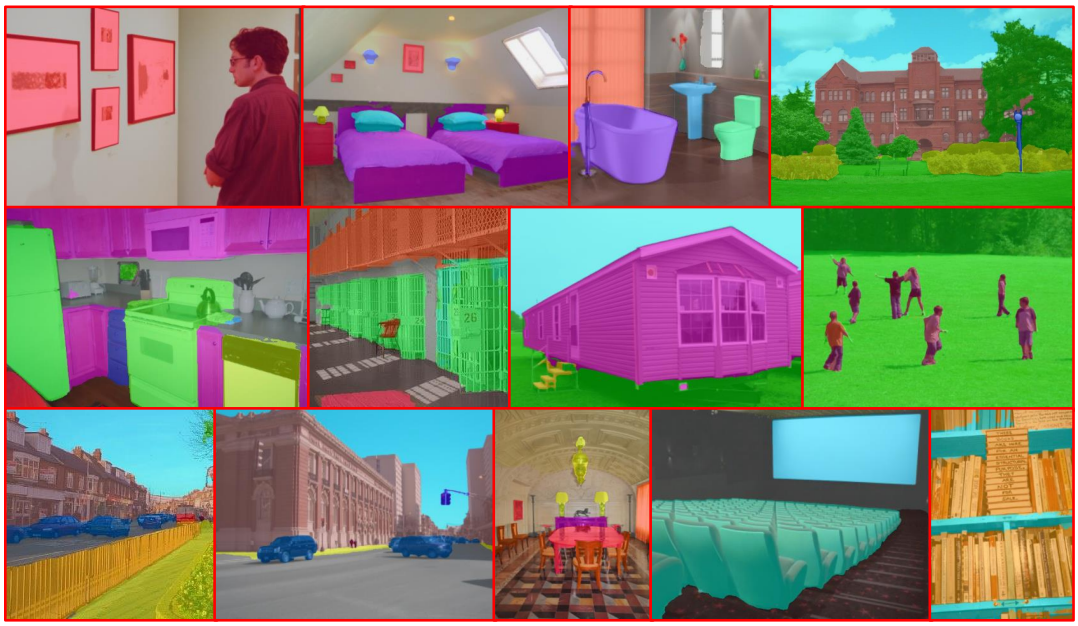


FIGURE 3.6. Competitive segmentation results on the ADE20K dataset. Our method performs well in various scenes and can capture detailed boundary information.

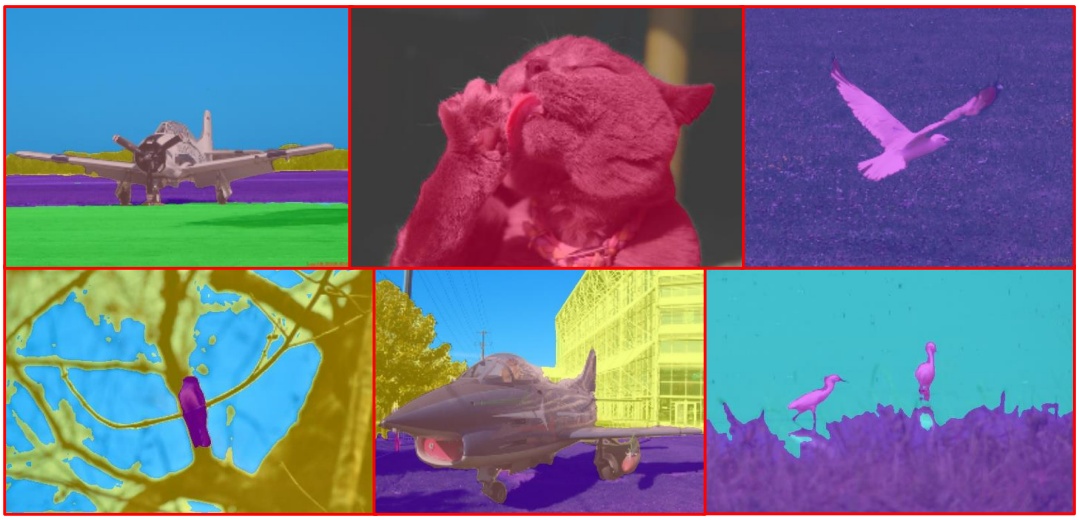


FIGURE 3.7. Competitive segmentation results on the PASCAL-Context dataset. The proposed method performs well on various shapes of objects.

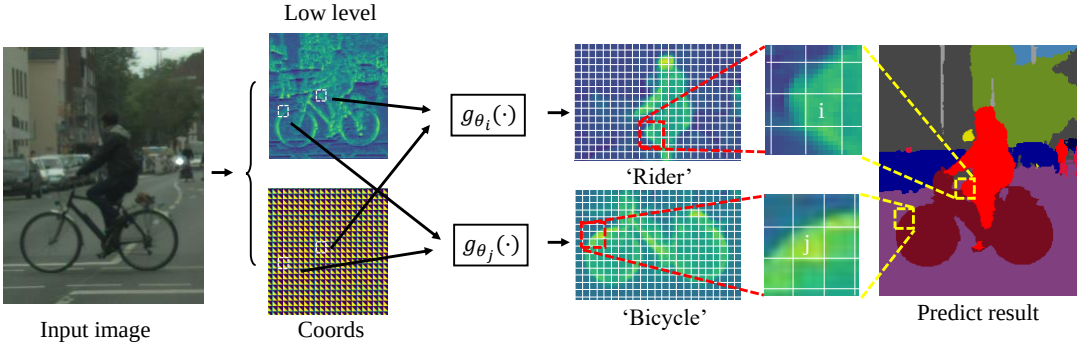


FIGURE 3.8. Detailed Illustration of the NRD module. Guidance maps from low-level feature maps and coordinate maps are concatenated together and pass through the representational networks $g_{\boldsymbol{\theta}}(\cdot)$.

Statement of Authorship

Title of Paper	SegViT: semantic segmentation with plain vision transformers
Publication Status	✓ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Published in NIPS 2022

Principal Author

Name of Principal Author (Candidate)	Bowen Zhang		
Contribution to the Paper	Proposed the ideas, conducted the experiments and write part of the paper		
Overall percentage (%)	70		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	10/8/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Zhi Tian		
Contribution to the Paper	Discussion, write the paper and the revision		
Signature		Date	17/08/2023

Name of Co-Author	Quan Tang		
Contribution to the Paper	Conduct some experiments		
Signature		Date	30/08/2023

Name of Co-Author	Xiangxiang Chu		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	17/08/2023

Name of Co-Author	Xiaolin Wei		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	17/08/2023

Name of Co-Author	Chunhua Shen		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	5/9/2023

Name of Co-Author	Yifan Liu		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	29/08/2023

Statement of Authorship

Title of Paper	SegViTv2: exploring efficient and continual semantic segmentation with plain vision transformers
Publication Status	☐ Published ☑ Accepted for Publication ✓ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Accepted in IJCV

Principal Author

Name of Principal Author (Candidate)	Bowen Zhang		
Contribution to the Paper	Proposed the ideas, conducted the experiments and write part of the paper		
Overall percentage (%)	70		
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.		
Signature		Date	10/8/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);

ii. permission is granted for the candidate in include the publication in the thesis; and

iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Liyang Liu		
Contribution to the Paper	Discussion, conduct some experiments and help with the writing		
Signature		Date	01/09/2023

Name of Co-Author	Vu Minh Hieu Phan		
Contribution to the Paper	Discussion, conduct some experiments and help with the writing		
Signature		Date	01/09/2023

Name of Co-Author	Zhi Tian		
Contribution to the Paper	Conduct some experiments		
Signature		Date	17/08/2023

Name of Co-Author	Chunhua Shen		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	5/9/2023

Name of Co-Author	Yifan Liu		
Contribution to the Paper	Discussion and write the revision		
Signature		Date	29/08/2023

Chapter 4

Segvit: Exploring Efficient and Continual Semantic Segmentation with Plain Vision Transformers

4.1 Introduction

Semantic segmentation, a pivotal task in computer vision, demands precise pixel-level classification of input images. Traditional methods, such as Fully Convolutional Networks (FCN) [Long et al., 2015a], which are widely used in state-of-the-art techniques, employ deep convolutional neural networks (ConvNet) as encoders or base models and a segmentation decoder to generate dense predictions. Prior works [Chen et al., 2018b; Wang et al., 2020; Yuan et al., 2020] have aimed to enhance performance by augmenting context information or incorporating multi-scale information, leveraging the inherent multi-scale and *hierarchical* attributes of the ConvNet architectures.

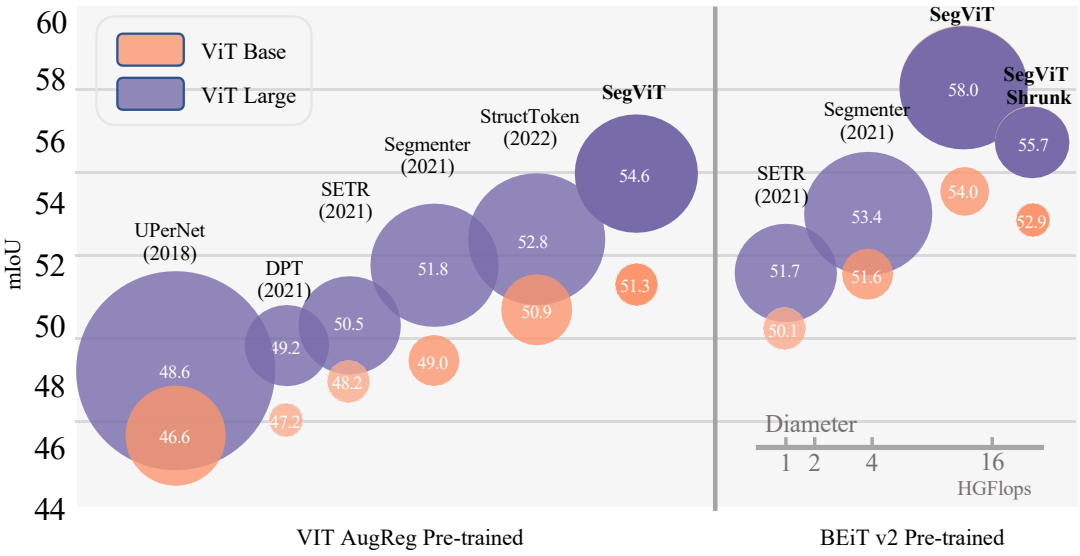


FIGURE 4.1. **Comparison with previous methods in terms of performance and efficiency** on ADE20K dataset. The orange and purple bubbles in the accompanying graph represent the ViT Base and ViT Large models, respectively, with the size of each bubble corresponding to the FLOPs of the variant segmentation methods. SegViT-BEiT Large achieves state-of-the-art performance with a **58.0%** mIoU on the ADE20K validation set. Additionally, our efficient, optimized version, SegViT-Shrunk-BEiT Large, saves half of the GFLOPs compared to UPerNet, significantly reducing computational overhead while maintaining a competitive performance of **55.7%**.

The advent of the Vision Transformer (ViT) [Dosovitskiy et al., 2021b] has offered a paradigm shift, serving as a robust backbone for numerous computer vision tasks. ViT, distinct from ConvNet base models, retains a plain and *non-hierarchical* architecture while preserving the resolution of the feature maps. To conveniently leverage existing segmentation decoders for dense prediction, recent Transformer-based approaches, including Swin Transformer [Liu et al., 2021] and PVT [Wang et al., 2021a], have developed *hierarchical* structures for feature representations. However, modifying the original ViT structures requires training the networks from scratch rather than using off-the-shelf plain ViT checkpoints that are pre-trained with large-scale datasets [Caron et al., 2021; Dehghani et al., 2023; Oquab et al., 2023]. Changing the plain ViT structure eliminates the potential for leveraging rich representations acquired from vision-language pre-training methods such as CLIP [Radford et al., 2021], BEiT [Bao et al., 2022], BEiT-v2 [Peng et al., 2022], MVP [Wei et al., 2022], and COTS [Lu et al., 2022]. Hence, there is a clear advantage to developing effective decoders for the original ViT structures in order to leverage those powerful representations. Previous works, such as UPerNet [Xiao et al., 2018] and DPT [Ranftl et al., 2021], have primarily focused on hierarchical feature maps and neglected the distinctive characteristics of the plain Vision Transformer. Consequently, these approaches result in computationally demanding operations with limited performance improvement, as illustrated in Figure 1. A recent trend in several works, such as SETR [Zheng et al., 2021a] or Segmenter [Strudel et al., 2021], aims to develop decoders specifically tailored for the Plain ViT architecture. However, these designs often represent a simplistic extension of per-pixel classification techniques derived from traditional convolution-based decoders. For example, SETR's decoder [Zheng et al., 2021a] uses a sequence of convolutions and bilinear up-sampling to increase the ViT's extracted feature maps gradually. It then applies a naive MLP to the extracted features to perform pixel-wise classification, which isolates the neighboring contexts surrounding the pixel. Current pixel-wise classification decoder designs overlook the importance of contextual learning when assigning labels to each pixel.

Another prevalent issue in deep networks, including Transformer, is 'catastrophic forgetting' [French, 1999; Kirkpatrick et al., 2017], where the model's performance on previously learned tasks deteriorates as it learns new ones [Phan et al., 2022; Shao and Feng, 2022; Wang et al., 2022c,d]. This limitation poses significant challenges for the application of deep segmentation models in dynamic real-world environments. Recently, the rapid development of the foundation model pre-trained on large-scale data has sparked interest among researchers in studying its transferability across various downstream tasks [Ostapenko et al., 2022]. These models are capable of extracting powerful and generalized representations, which has led to a growing interest in exploring their extensibility to new classes and tasks while retaining the previously learned knowledge representations [Ramasesh et al., 2022; Wu et al., 2022a].

Motivated by these challenges, this chapter aims to explore how a plain vision transformer can perform semantic segmentation tasks more effectively without the need for a hierarchical backbone redesign. As self-supervision and multi-modality pre-training continue to evolve, we anticipate that the plain vision transformer will learn enhanced visual representations. Consequently, decoders for dense tasks are expected to adapt more flexibly and efficiently to these representations.

In light of these research gaps, we propose **SegViT** — a novel, efficient segmentation network that features a plain Vision Transformer and exhibits robustness against forgetting. We introduce a novel Attention-to-Mask (ATM) module that operates as a lightweight component for the SegViT decoder. Leveraging the non-linearity of cross-attention learning, our proposed ATM employs learnable class tokens as queries

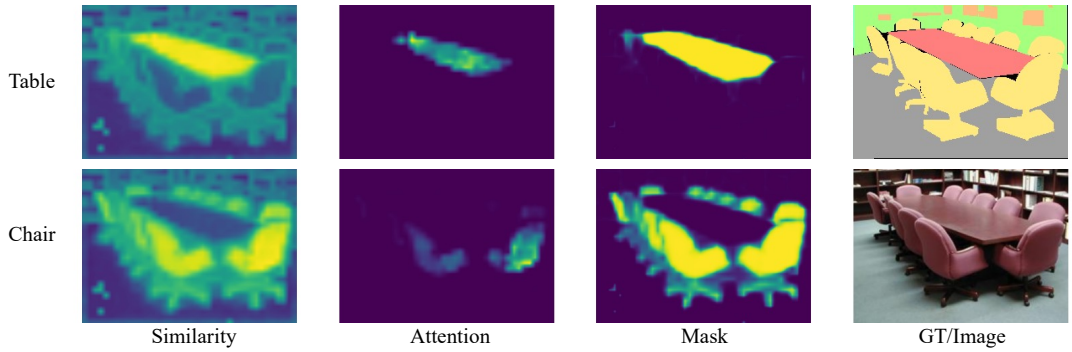


FIGURE 4.2. **The overall concept of our Attention-to-Mask decoder.** ATM learns the similarity map for each category by capturing the cross-attention between the class tokens and the spatial feature map (Left). `Sigmoid` is applied to produce category-specific masks, highlighting the area with high similarity to the corresponding class (Middle). ATM enhances the semantic representations by encouraging the feature to be similar to the target class token and dissimilar to other tokens.

to pinpoint spatial locations that exhibit high compatibility with each class. We advocate for regions affiliated with a particular class to possess substantial similarity values that correspond to the respective class token.

As depicted in Figure. 4.2, the ATM generates a meaningful similarity map that accentuates regions with a strong affinity towards the 'Table' and 'Chair' categories. By simply implementing a `Sigmoid` operation, we can transform these similarity maps into mask-level predictions. The computation of the mask scales linearly with the number of pixels, a negligible cost that can be integrated into any backbone to bolster segmentation accuracy. Building upon this efficient ATM module, we present a novel semantic segmentation paradigm that utilizes the cost-effective structure of plain ViT, referred to as SegViT. Within this paradigm, multiple ATM modules are deployed at various layers to extract segmentation masks at different scales. The final prediction is the summation of the outputs derived from these layers.

To alleviate the computational burdens of plain Vision Transformers (ViTs), we introduce the 'Shrunk' structure, which incorporates query-based downsampling (QD) and query-based upsampling (QU). The proposed QD employs a 2x2 nearest neighbor downsampling technique to obtain a sparser token mesh, reducing the number of tokens involved in attention computations. Moreover, we extend QD to edge-aware query-based downsampling (EQD). EQD selectively preserves tokens situated at object edges, as they possess more discriminative information. Consequently, QU recovers the discarded tokens within the object's homogeneous body, reconstructing high-resolution features crucial for accurately dense prediction. By integrating the 'Shrunk' structure with the ATM module as the decoder, we achieve computational reductions of up to 50% while maintaining competitive performance levels.

We further extend the application of our SegViT framework to continual learning. With the powerful and generalized representation that the foundation model has learned, this chapter aims to study the ability to extend the foundation model to new classes and new tasks without forgetting the knowledge it has learned. Recent techniques in continual semantic segmentation (CSS) aim to replay old data [Cha et al., 2021; Maracani et al., 2021] or distill knowledge from the previous model to mitigate model divergence [Cermelli et al., 2020; Phan et al., 2022; Zhang et al., 2022b]. These methods necessitate fine-tuning parameters responsible for old tasks, which can disrupt the previously learned solutions, leading to forgetting. In contrast, our proposed

SegViT supports learning new classes without encroaching on previously acquired knowledge. We strive to establish a forget-free SegViT framework, which incorporates a new ATM module dedicated to new tasks while keeping all old parameters in a frozen state. Consequently, the proposed SegViT has the potential to virtually eliminate the issue of forgetting.

Our key contributions can be summarized as follows:

- We introduce the Attention-to-Mask (ATM) decoder module, a potent and efficient tool for semantic segmentation. For the first time, we exploit spatial information present in attention maps to generate mask predictions for each category, proposing a new paradigm for semantic segmentation.
- We present the *Shrunk* structure, applicable to any plain ViT backbone, which alleviates the intrinsically high computational expense of the *non-hierarchical* ViT while maintaining competitive performance, as illustrated in Figure. 4.1. We are the inaugural work capitalizing on edge information to decrease and restore tokens for efficient computation. Our *Shrunk* version of SegViT, tested on the ADE20K dataset, achieves a mIoU of 55.7%, with a computational cost of 308.8 GFLOPs, marking a reduction of approximately 50% compared to the original SegViT (637.9 GFLOPs).
- We propose a new SegViT architecture capable of continual learning devoid of forgetting. To our knowledge, we are the first work to seek to completely freeze all parameters for old classes, thereby nearly obliterating the issue of catastrophic forgetting.

4.2 Our Methods

In this section, we will first introduce the overall architecture of our proposed SegViT model. Then, we will proceed to discuss the 'Shrunk' architecture, which is designed to reduce the overall computational cost of the model. Additionally, we will delve into the continuous semantic segmentation setting and adapt our SegViT model framework to seamlessly align with this setting.

4.2.1 Overall SegViT architecture

SegViT comprises a ViT-based encoder responsible for feature extraction and a decoder used to learn the segmentation map. In terms of the encoder, we have designed the 'Shrunk' structure to decrease the computational costs associated with the plain ViT. As for the decoder, we introduce a novel lightweight module called Attention-to-Mask (ATM). This module generates class-specific masks denoted as M and class predictions denoted as P, which determine the presence of a particular class in the image. The mask outputs from a stack of ATM modules are combined and then multiplied with the class predictions to obtain the final segmentation output. Figure. 4.3 illustrates the overall architecture of our proposed SegViT.

4.2.1.1 Encoder

Given an input image $I \in \mathbb{R}^{H\times W\times 3}$, the plain vision transformer backbone reshapes it into a sequence of tokens $\mathcal{F}_0 \in \mathbb{R}^{L\times C}$, where $L = \frac{HW}{P^2}$, P is the patch size, and C is the number of channels. To capture positional information, learnable position embeddings of the same size as $\mathcal{F}_0$ are added. Subsequently, the token sequence $\mathcal{F}_0$

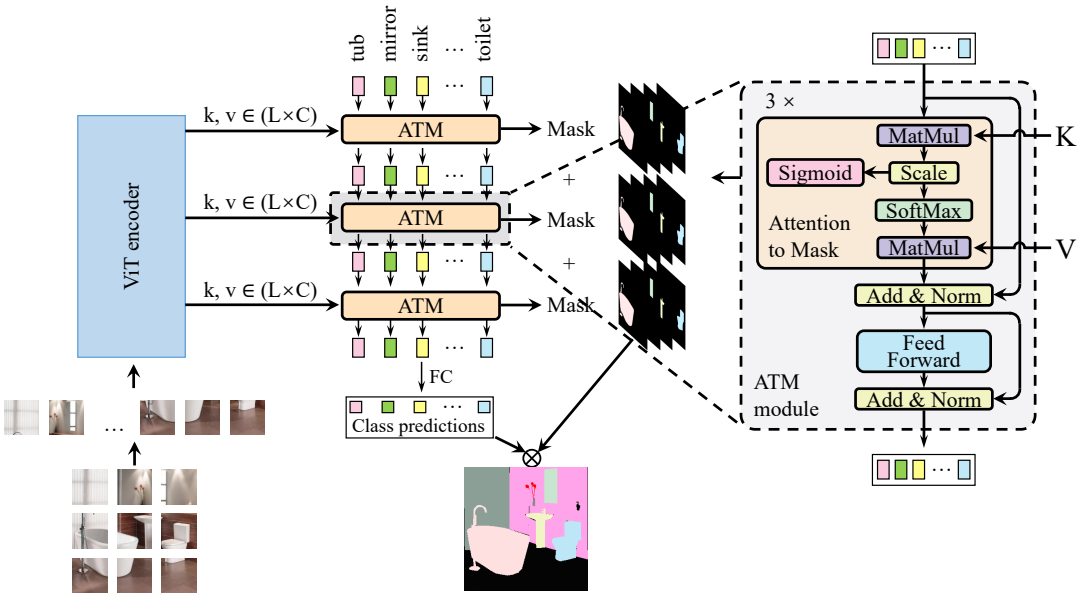


FIGURE 4.3. **The overall SegViT structure with the ATM module.** The Attention-to-Mask (ATM) module inherits the typical transformer decoder structure. It takes in randomly initialized class embeddings as queries and the feature maps from the ViT backbone to generate keys and values. The outputs of the ATM module are used as the input queries for the next layer. The ATM module is carried out sequentially with inputs from different layers of the backbone as keys and values in a cascade manner. A linear transform is then applied to the output of the ATM module to produce the class predictions for each token. The mask for the corresponding class is transferred from the similarities between queries and keys in the ATM module. We have removed the self-attention mechanism in ATM decoder layers further improve the efficiency while maintaining the performance.

undergoes m transformer layers to produce the output. The output tokens for each layer are defined as $[\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_m] \in \mathbb{R}^{L\times C}$.

In the case of a plain vision transformer such as ViT, there are no additional modules involved, and the number of tokens remains unchanged for each layer. However, the computational costs of plain ViT can be prohibitively expensive. To address this issue, we introduce the *Shrunk* structure, which enables the development of an efficient ViT-based encoder. Further details regarding the *Shrunk* structure can be found in Section 4.2.2.

4.2.1.2 Decoder

Attention-to-Mask (ATM).

Cross-attention can be described as the mapping between two sequences of tokens, denoted as $\{\mathbf{v_1}, \mathbf{v_2}\}$. In our case, we define two token sequences: $\mathcal{G} \in \mathbb{R}^{N\times C}$ with a length N equal to the number of classes, and $\mathcal{F}_i \in \mathbb{R}^{L\times C}$. To enable cross-attention, linear transformations are applied to each token sequence, resulting in the query (Q), key (K), and value (V) representations. This process is described by Equation (4.1).

$$\begin{aligned} Q &= \phi_q(\mathcal{G}) \in \mathbb{R}^{N\times C}, \\ K &= \phi_k(\mathcal{F}_i) \in \mathbb{R}^{L\times C}, \\ V &= \phi_v(\mathcal{F}_i) \in \mathbb{R}^{L\times C}. \end{aligned} \tag{4.1}$$

The similarity map is calculated by computing the dot product between the query and key representations. Following the scaled dot-product attention mechanism, the similarity map and attention map are calculated as follows:

$$\begin{aligned} S(Q,K) &= \frac{QK^T}{\sqrt{d_k}} \in \mathbb{R}^{N\times L}, \\ Attention(\mathcal{G},\mathcal{F}_i) &= \texttt{Softmax}\big(S(Q,K)\big)V \in \mathbb{R}^{N\times C}, \end{aligned} \tag{4.2}$$

where $\sqrt{d_k}$ is a scaling factor with d_k equals to the dimension of the keys. The shape of the similarity map $S(Q,K)$ is determined by the lengths of the two token sequences, N and L. The attention mechanism updates $\mathcal{G}$ by performing a weighted sum of V, where the weights are derived from the similarity map after applying the softmax function along the L dimension.

In dot-product attention, the softmax function is used to concentrate attention exclusively on the token with the highest similarity. However, we believe that tokens other than those with maximum similarity also carry meaningful information. Based on this intuition, we have designed a lightweight module that generates semantic predictions more directly. To achieve this, we assign $\mathcal{G}$ as the class embeddings for the segmentation task and $\mathcal{F}_i$ as the output of layer i of the ViT backbone. A semantic mask is paired with each token in $\mathcal{G}$ to represent the semantic prediction for each class. The binary mask M is defined as follows:

$$Mask(\mathcal{G},\mathcal{F}_i) = \texttt{Sigmoid}(S(Q,K)) \in \mathbb{R}^{N\times L}. \tag{4.3}$$

The masks have a shape of $N \times L$, which can be reshaped to $N \times \frac{H}{P} \times \frac{W}{P}$ and bilinearly upsampled to the original image size $N \times H \times W$. The ATM mechanism, as illustrated in the right part of Figure 4.3, produces masks as its intermediate output during the cross-attention process.

The final output tokens $Z \in \mathbb{R}^{L\times C}$ from the ATM module are utilized for classification. A fully connected layer (FC) parameterized by $W \in \mathbb{R}^{C\times 2}$ followed by the Softmax function is used to predict whether the object class is present in the image or not. The class predictions $\mathcal{P} \in \mathbb{R}^{N\times 2}$ are formally defined as:

$$\mathcal{P} = \text{Softmax}(WZ). \tag{4.4}$$

Here, $P_{c,1}$ indicates the likelihood of class c appearing in the image. For simplicity, we refer to P_c as the probability score for class c.

The output segmentation map for class $O_s \in \mathbb{R}^{H\times W}$ is obtained by element-wise multiplication of the reshaped class-specific mask M_c and its corresponding prediction score P_c: $O_c = P_c \odot M_c$. During inference, the label is assigned to each pixel i by selecting the class with the highest score using $\text{argmax}_c O_{i,c}$.

Indeed, plain base models like ViT do not inherently possess multiple stages with features of different scales. Consequently, structures such as Feature Pyramid Networks (FPN) that merge features from multiple scales are not applicable to them.

Nevertheless, features from layers other than the last one in ViT contain valuable low-level semantic information, which can contribute to improving performance. In SegViT, we have developed a structure that leverages feature maps from different layers of ViT to enrich the feature representations. This allows us to incorporate and benefit from the rich low-level semantic information present in those feature maps.

SegViT is trained via the classification loss and the binary mask loss. The classification loss ($\mathcal{L}_{\text{cls}}$) minimizes cross-entropy between the class prediction and the actual

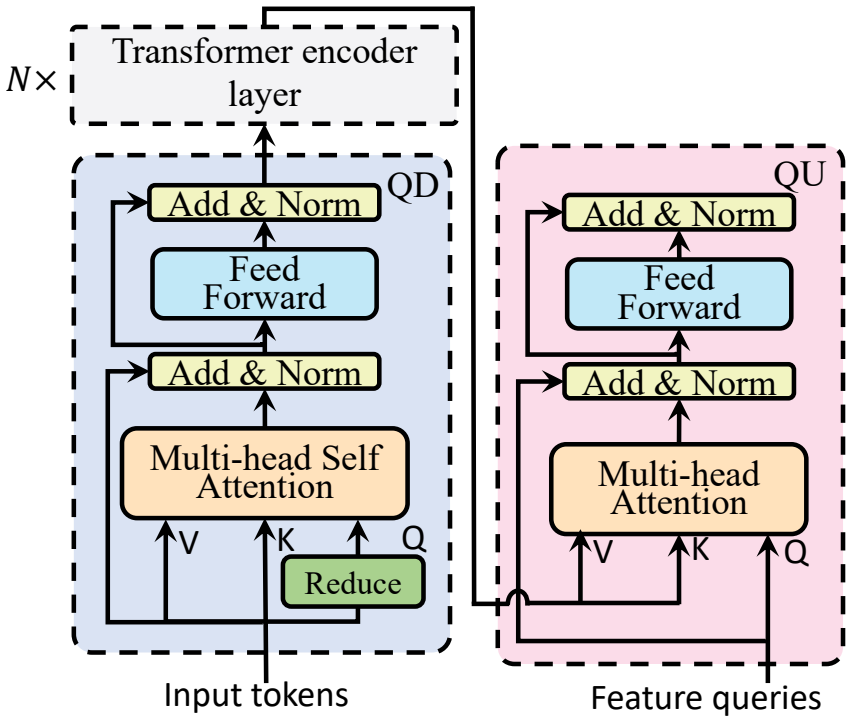


FIGURE 4.4. Architecture of the proposed query-downsapling (QD) layer (**blue** block) and the query-upsampling (QU) layer (pink block). The QD layer uses an efficient down-sampling technique (green block) and removes less informative input tokens used for the query. The QU layer takes a set of trainable query tokens and learns to recover the discarded tokens using multi-head attention.

target. The mask loss ($\mathcal{L}_{\text{mask}}$) consists of a focal loss [Lin et al., 2017c] and a dice loss [Milletari et al., 2016] for optimizing the segmentation accuracy and addressing sample imbalance issues in mask prediction. The dice loss and focal loss respectively minimize the dice and focal scores between the predicted masks and the ground-truth segmentation. The final loss is the combination of each loss, formally defined as:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda_{\text{focal}}\mathcal{L}_{\text{focal}} + \lambda_{\text{dice}}\mathcal{L}_{\text{dice}} \tag{4.5}$$

where λ_{focal} and λ_{dice} are hyperparameters that control the strength of each loss function. Previous mask transformer methods such as MaskFormer [Cheng et al., 2021b] and DETR [Carion et al., 2020b] have adopted the binary mask loss and fine-tuned their hyperparameters through empirical experiments. Hence, for consistency, we directly use the same values as MaskFormer and DETR for the loss hyperparameters: $\lambda_{\text{focal}} = 20.0$ and $\lambda_{\text{dice}} = 1.0$.

4.2.2 *Shrunk* Structure for Efficient Plain ViT Encoder

Recent efforts, such as DynamicViT [Rao et al., 2021], TokenLearner [Ryoo et al., 2021], and SPViT [Kong et al., 2022], propose token pruning techniques to accelerate vision transformers. However, most of these approaches are specifically designed for image classification tasks and, as a result, discard valuable information. When these techniques are applied to semantic segmentation, they may fail to preserve high-resolution features that are necessary for accurate dense prediction tasks.

In this chapter, we propose the *Shrunk* structure, which utilizes query-based down-sampling (QD) to prune the input token sequence $\mathcal{F}_i$, and query up-sampling (QU) to recover the discarded tokens, thereby preserving the fine-detailed features that are crucial for semantic segmentation. The overall architecture of QD and QU is illustrated in Figure 4.4.

For QD, we have re-designed the Transformer encoder block [Vaswani et al., 2017a] and incorporated efficient down-sampling operations to specifically reduce the number of query tokens. In a Transformer encoder layer, the computational cost is directly influenced by the number of query tokens, and the output size is determined by the query token size. To mitigate the computational burden while maintaining information integrity, a feasible strategy is to selectively reduce the number of query tokens while preserving the key and value tokens. This approach allows for an effective reduction in the output size of the current layer, leading to reduced computational costs for subsequent layers.

For QU, we achieve upsampling by employing either a pre-defined or inherited token sequence with a higher resolution as the query tokens. The key and value tokens are taken from the token sequence obtained from the backbone, which typically has a lower resolution. The output size is dictated by the query tokens with higher resolution. Through the cross-attention mechanism, information from the key and value tokens is integrated into the output. This process facilitates a non-linear merging of information and demonstrates an upsampling behavior, effectively increasing the resolution of the output.

Our proposed *Shrunk* structure incorporates the QD and QU modules. Specifically, we integrate a QD operation at the middle depth of the ViT backbone, precisely at the 8^{th} layer of a 24-layer backbone. The QD operation downsamples the query tokens using a 2×2 nearest neighbor downsampling operation, resulting in a feature map size reduction from $1/16$ to $1/32$. However, such downsampling can potentially cause information loss and performance degradation. To mitigate this issue, prior to applying the QD operation, we employ a QU operation to the feature map. This involves initializing a set of query tokens with a resolution of $1/16$ to store the information. Subsequently, as the downsampled feature map progresses through the remaining backbone layers, it is merged and upsampled using another QU operation alongside the previously stored $1/16$ high-resolution feature map. This iterative process ultimately generates a $1/16$ high-resolution feature map enriched with semantic information processed by the backbone.

However, the direct application of query downsampling (QD) during the initial stage of the encoder leads to a decline in performance. Shallow layers, responsible for capturing low-level features before global attention updates, are highly sensitive to nearest downsampling, resulting in significant information loss. Notably, critical information, especially tokens containing multiple categories (decision boundaries), can unpredictably disappear. To address this limitation, we introduce the Edged Query Downsampling (EQD) technique to refine the QD process. In EQD, we employ a 2×2 sparse downsampling while also retaining tokens containing multiple categories (tokens with edges). By preserving the 2×2 sparse tokens, we ensure the retention of essential semantic information, and by keeping the edge tokens, we retain detailed spatial information. This dual preservation strategy effectively minimizes the loss of valuable data and successfully overcomes the limitations associated with low-level layers, resulting in improved performance and enriched representations throughout the encoder structure.

To extract edges, we add a separate branch using a lightweight multilayer perceptron (MLP) edge detection head that learns to detect edges from the input image.

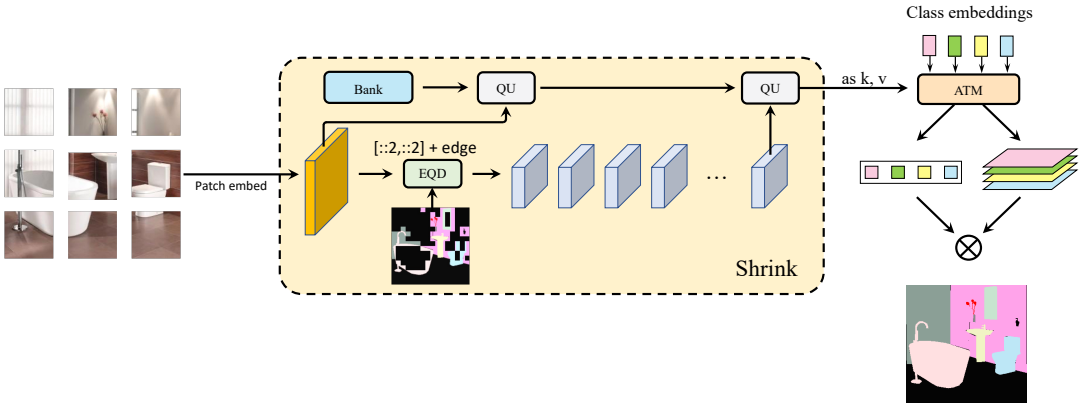


FIGURE 4.5. **Illustrations of the *Shrink* structure.** The provided diagram showcases the blue transformer encoder block and the orange patch embedding block. In our innovative *Shrunk* structure, we strategically implement query downsampling (QD) immediately after the patch embedding step to capture crucial information effectively. To further maximize information retention compared to the basic nearest downsampling approach used by QD, we introduce the Edged Query Downsampling (EQD) technique. The EQD technique consolidates four adjacent tokens into one token while also incorporating tokens containing edges, which significantly enhances the downsampling process. Notably, the EQD ensures that tokens preserving rich semantic information and decision boundaries are retained. Additionally, we utilize a lightweight parallel edge detection head to extract edge information. By skillfully incorporating these logical improvements, the *Shrunk* structure achieves enhanced efficiency while retaining vital information, making it exceptionally well-suited for various applications that demand both speed and accurate representation.

The edge detection head operates as an auxiliary branch, trained simultaneously with the main ATM decoder. This head processes the input image, which has the same dimensions as the backbone. Let the input image have C channels, aligned with the backbone. The Multi-Layer Perceptron (MLP) in this head consists of three layers, with dimensions C, $C/2$, and 2, respectively. Let I represent the input image, and the output of the MLP can be defined as $E = \text{MLP}(I; W_1, W_2, W_3)$, where W_1, W_2, W_3 are the weights for the three layers. The output E is then passed through a softmax activation function, resulting in $S = \text{Softmax}(E)$. To determine the confidence level of a token belonging to an edge, we apply a threshold τ. In our implementation, we set τ to 0.7. To obtain the ground-truth (GT) edge, we perform post-processing on the GT segmentation map Y. Since the input has been tokenized with a patch size of P, we tokenize the GT and reshape it into a sequence of tokens denoted as $Y \in R^{(HW/P^2)\times P\times P}$, where the last two dimensions correspond to the patch dimensions. We consider a patch to contain an edge if there exists any edge pixel within the patch. We define the edge mask $Mask_i$ as follows:

$$Mask_i = \begin{cases} 1 & \text{if } \sum_{j,k} Y_{i,j,k} > 0, \\ 0 & \text{otherwise.} \end{cases} \tag{4.6}$$

For each element s_i in S, we create a binary edge mask M_i: $M_i = 1, \text{if } s_i \geq \tau$. The cross-entropy loss is computed between the generated edge mask M_i and the ground-truth edge mask Y_i: $\mathcal{L}_{\text{edge}} = -\sum iY_i \log(M_i) + (1 - Y_i) \log(1 - M_i)$. By incorporating the Edge Detection head as an auxiliary branch, the Shrunk architecture effectively retains detailed spatial contexts throughout the query downsampling process, forming an Edge Query Downsampling (EQD) structure. This EQD structure allows the model to capture and preserve edge information during spare downsampling, resulting in a

significant reduction in computational overhead without compromising performance. The integration of EQD enables the Shrunk architecture to strike a remarkable balance between computational efficiency and maintaining high-performance levels.

4.2.3 Exploration on Continual Semantic Segmentation

Continual semantic segmentation aims to train a segmentation model in T steps without forgetting. At step t, we are given a dataset $\mathcal{D}^t$ which comprises a set of pairs (X^t, Y^t), where X^t is an image of size $H \times W$ and Y^t is the ground-truth segmentation map.

Here, Y^t only consists of labels in current classes $\mathcal{C}^t$, while all other classes (i.e., old classes $\mathcal{C}^{1:t-1}$ or future classes $\mathcal{C}^{t+1:T}$) are assigned to the background. In continual learning, the model at step t should be able to predict all classes $\mathcal{C}^{1:t}$ in history.

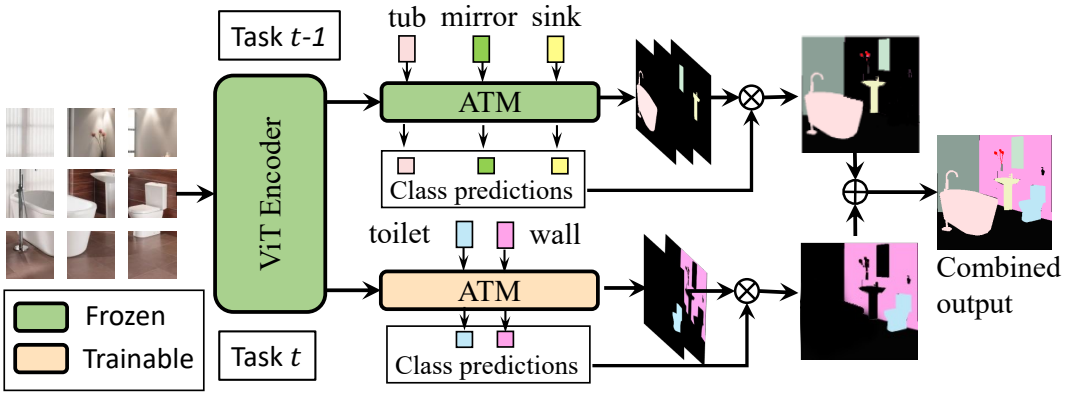


FIGURE 4.6. Overview of SegViT adapted for continual semantic segmentation. When learning a new task t, we grow and train a separate ATM and fully-connected layer to produce mask and class prediction. All the parameters dedicated to the old task $t-1$, including ATM, FC layer, and the ViT encoder, are frozen. This prevents interfering with the old knowledge, which guarantees no forgetting.

SegViT for Continual Learning. Existing continual semantic segmentation methods [Phan et al., 2022; Zhang et al., 2022b] propose regularization algorithms to preserve the past knowledge of a specific architecture, DeepLabV3. These methods focus on continual semantic segmentation for DeepLabV3 with a ResNet backbone, which has a less robust visual representation for distinguishing between different categories. Consequently, these methods require fine-tuning model parameters to learn new classes while attempting to retain knowledge of old classes. Unfortunately, adapting the old parameters dedicated to the previous task inevitably interferes with past knowledge, leading to catastrophic forgetting. In contrast, our method, SegViT, decouples class prediction from mask segmentation, making it inherently suitable for a continual learning setting. By leveraging the powerful representation capability of the plain vision transformer, we can learn new classes by solely fine-tuning the class proxy (i.e., the class token) while keeping the old parameters frozen. This approach eliminates the need for fine-tuning old parameters when learning new tasks, effectively addressing the issue of catastrophic forgetting.

When training on a current task t, we add a new sequence of learnable tokens $\mathcal{G}^t \in \mathbb{R}^{|\mathcal{C}^t| \times C}$ with the length equals to the number of classes $|\mathcal{C}^t|$ in the current task. To learn new classes, we grow and train new ATM modules and a fully-connected layer for mask prediction and mask classification. For simplicity, we ignore the parallel structure of ATM modules. A single ATM module refers to multiple ATM modules.

Let A^t and W^t denote the ATM module and the weights of the fully connected (FC) layer for task t. All parameters for a previous task, including the ViT encoder, the ATM module, and the FC layer, are completely frozen. Fig. 4.6 illustrates the overview of our SegViT architecture adapted for continual semantic segmentation.

Given the encoder extracted features $\mathcal{F}_T$ and the class tokens $\mathcal{G}^t$, the ATM produces the mask predictions M^t and the output tokens Z^t corresponding to the mask:

$$M^t, Z^t = \text{ATM}(\mathcal{G}^t, \mathcal{F}^T). \tag{4.7}$$

Based on Eq. 4.4, the class prediction $\mathcal{P}$ is obtained by applying FC on the class token Z^t. The prediction score for each class S_c^t is multiplied with the mask M_c^t to get the segmentation map O_c^t for class c:

$$O_c^t = S_c^t \odot M_c^t, \tag{4.8}$$

where $\odot$ denotes the element-wise multiplication.

The segmentation $\hat{O}^t$ is obtained by taking the class c having the highest score in every pixel, defined as

$$\hat{O}^t = \underset{c \in \mathcal{C}^t}{\text{argmax}}\ O_{i,c}^t \tag{4.9}$$

Based on the ground truth Y^t for task t, SegViT is trained using the loss function defined in Eq. 4.5. To produce the final segmentation for all tasks, we concatenate the outputs O^t from all tasks.

4.3 Experiments

4.3.1 Datasets

ADE20K [Zhou et al., 2017a] is a challenging scene parsing dataset which contains 20,210 images as the training set and 2,000 images as the validation set with 150 semantic classes.

COCO-Stuff-10K [Caesar et al., 2018b] is a scene parsing benchmark with 9,000 training images and 1,000 test images. Even though the dataset contains 182 categories, not all categories exist in the test split. We follow the implementation of mmsegmentation [MMSegmentation, 2020] with 171 categories to conduct the experiments.

PASCAL-Context [Mottaghi et al., 2014b] is a dataset with 4,996 images in the training set and 5,104 images in the validation set. There are 60 semantic classes in total, including a class representing 'background'.

4.3.2 Implementation details

Transformer backbone. We employ the naive ViT [Dosovitskiy et al., 2021b] as the backbone for our method. Specifically, we utilize the 'Base' variation of ViT for most of our ablation studies and provide results based on the 'Large' variation as well. It is worth noting that different pre-trained weights can lead to significant variations in performance, as suggested by Segmenter [Strudel et al., 2021]. Therefore, to ensure a fair comparison, we adopt the pre-trained weights provided by Augreg [Steiner et al., 2021], following the practices of counterparts such as Strudel [Strudel et al., 2021] and StructToken [Lin et al., 2022]. These weights are obtained through training on ImageNet-21k with strong data augmentation and regularization techniques [Steiner et al., 2021]. To explore the maximum capacity and assess the upper bound of our method, we also

TABLE 4.1. Experiment results on the ADE20K `val.` split. 'ms' means that mIoU is calculated using multi-scale inference. '†' means the models use the backbone weights pre-trained by AugReg [Steiner et al., 2021]. '*' represents the model reproduced under the same settings as the official repo. The GFLOPs are measured at single-scale inference with the given crop size. We report inference speed for our SegViT and reproduce previous methods in terms of Frame Per Second (FPS) on a single A100 device.

Method	Backbone	Crop Size	GFLOPs	mIoU (ss)	mIoU (ms)	Inf time (fps)
UPerNet [Xiao et al., 2018]	ViT-Base	512 × 512	443.9	46.6	47.5	16.07
DPT* [Ranftl et al., 2021]	ViT-Base	512 × 512	219.8	47.2	47.9	23.63
SETR-MLA* [Zheng et al., 2021a]	ViT-Base	512 × 512	113.5	48.2	49.3	-
Segmenter* [Strudel et al., 2021]	ViT-Base	512 × 512	129.6	49.0	50.0	20.46
StructToken [Lin et al., 2022]	ViT-Base	512 × 512	171.5	50.9	51.8	14.22
MaskFormer [Cheng et al., 2021c]	Swin-B(21K)	640 × 640	198.3	52.7	53.9	-
Mask2Former [Cheng et al., 2022a]	Swin-B(21K)	640 × 640	223.4	*53.9*	**55.1**	12.43
SegViT (Ours)	ViT-Base	512 × 512	120.9	51.3	53.0	31.52
SegViT (*Shrunk*, Ours)	BEiTv2-Base	512 × 512	**74.4**	52.9	53.3	25.03
SegViT (Ours)	BEiTv2-Base	512 × 512	120.9	**54.0**	*54.9*	23.59
DPT* [Ranftl et al., 2021]	ViT-Large†	640 × 640	800.0	49.2	49.5	9.38
UPerNet [Xiao et al., 2018]	ViT-Large†	640 × 640	1993.9	48.6	50.0	3.88
SETR-MLA [Zheng et al., 2021a]	ViT-Large	512 × 512	368.6	48.6	50.3	5.17
MCIBI [Jin et al., 2021a]	ViT-Large	512 × 512	>400	-	50.8	-
Segmenter [Strudel et al., 2021]	ViT-Large†	640 × 640	671.8	51.8	53.6	4.73
StructToken [Lin et al., 2022]	ViT-Large†	640 × 640	774.6	52.8	54.2	4.1
KNet+UPerNet [Zhang et al., 2021b]	Swin-L(21K)	640 × 640	659.3	52.2	53.3	11.28
MaskFormer [Cheng et al., 2021c]	Swin-L(21K)	640 × 640	378.1	54.1	55.6	10.21
Mask2Former [Cheng et al., 2022a]	Swin-L(21K)	640 × 640	402.7	56.1	57.3	8.81
SegViT (ours)	ViT-Large†	640 × 640	637.9	54.6	55.2	9.37
SegViT(*Shrunk* , ours)	ViT-Large†	640 × 640	209.1	53.0	54.9	10.26
SegViT (*Shrunk*, ours)	BEiTv2-Large†	512 × 512	210.3	55.1	56.1	9.82
SegViT (ours)	BEiTv2-Large†	512 × 512	374.0	*56.5*	*58.0*	9.39
SegViT (*Shrunk*, ours)	BEiTv2-Large†	640 × 640	308.8	55.7	57.0	9.38
SegViT (ours)	BEiTv2-Large†	640 × 640	637.9	**58.0**	**58.2**	6.25

conduct experiments using stronger base models such as DEiT v3 [Touvron et al., 2022] and BEiT v2 [Peng et al., 2022].

Training settings. We use MMSegmentation [MMSegmentation, 2020] and follow the commonly used training settings. During training, we applied data augmentation sequentially via random horizontal flipping, random resize with the ratio between 0.5 and 2.0, and random cropping (512 × 512 for all except that we use 480 × 480 for PASCAL-Context and 640 × 640 for ViT-large on ADE20K). The batch size is 16 for all datasets with a total iteration of 160k, 80k, and 80k for ADE20k, COCO-Stuff-10k, and PASCAL-Context respectively.

Evaluation metric. We use the mean Intersection over Union (mIoU) as the metric to evaluate the performance. 'ss' means single-scale testing and 'ms' test time augmentation with multi-scaled $(0.5, 0.75, 1.0, 1.25, 1.5, 1.75)$ inputs. All reported mIoU scores are in a percentage format. All reported computational costs in GFLOPs are measured using the fvcore[1] library.

4.3.3 Comparisons with the State-of-the-art Methods

Results on ADE20K.

Table. 4.1 reports the comparison with the state-of-the-art methods on ADE20K validation set using ViT backbone. The SegViT uses the ATM module with multi-layer inputs from the original ViT backbone, while the *Shrunk* is the one that conducts QD to the ViT backbone and saves 50% of the computational cost without sacrificing

[1] https://github.com/facebookresearch/fvcore

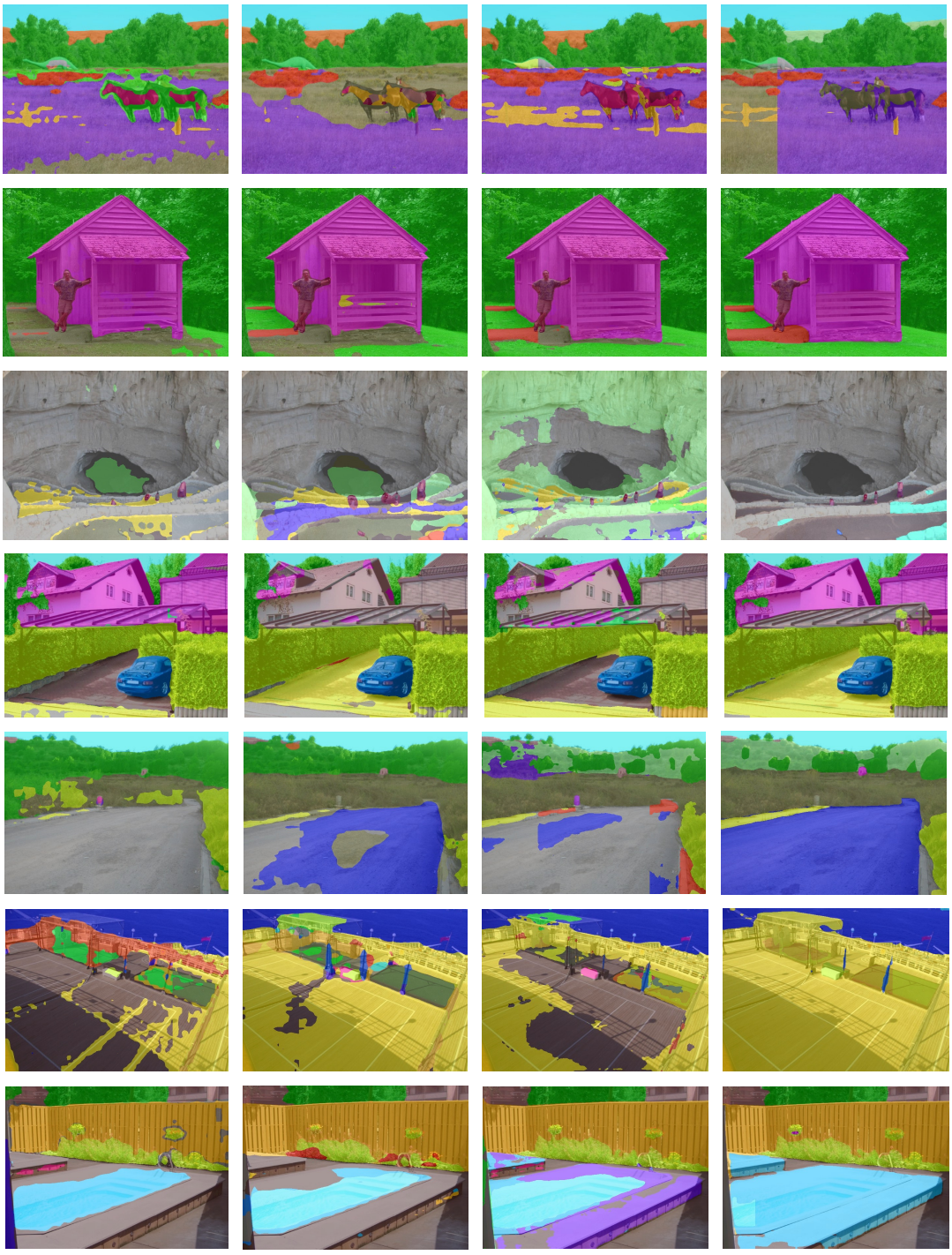
TABLE 4.2. Experiment results on the COCO-Stuff-10K `test.` split. Following published methods, we report the results with multi-scale inference (denoted by 'ms'). The GFLOPs is measured at single scale inference with a crop size of 512 × 512.

Method	Backbone	GFLOPs	mIoU (ms)
DANet [Fu et al., 2019a]	Dilated-ResNet-101	289.3	39.7
MaskFormer [Cheng et al., 2021b]	ResNet-101-fpn	81.7	39.8
EMANet [Li et al., 2019]	Dilated-ResNet-101	247.4	39.9
SpyGR [Li et al., 2020a]	ResNet-101-fpn	>80	39.9
OCRNet [Yuan et al., 2020]	HRNetV2-W48	167.9	40.5
GINet [Wu et al., 2020]	JPU-ResNet-101	>200	40.6
RecoNet [Chen et al., 2020]	Dilated-ResNet-101	>200	41.5
ISNet [Jin et al., 2021b]	Dilated-ResNeSt-101	228.3	42.1
MCIBI [Jin et al., 2021a]	ViT-Large	>380	44.9
StructToken [Lin et al., 2022]	ViT-Large	>400	49.1
SenFormer [Bousselham et al., 2021]	Swin-Large	>400	50.1
SegViT (*Shrunk*, ours)	ViT-Large	224.8	49.4
SegViT (ours)	ViT-Large	383.9	50.3
SegViT (*Shrunk*, ours)	BEiTv2-Large	**213.3**	50.54
SegViT (ours)	BEiTv2-Large	388.2	**53.46**

TABLE 4.3. Experimental results on the PASCAL-Context `val.` split. Following published methods, we report the results with multi-scale inference (denoted by 'ms'). mIoU_{59}: mIoU averaged over 59 classes (without background). mIoU_{60}: mIoU averaged over 60 classes (59 classes plus background). Both metrics were used in the literature, and we report for the 60 classes. The GFLOPs are measured at single scale inference with a crop size of 480 × 480.

Method	Backbone	GFLOPs	mIoU_{59} (ms)	mIoU_{60} (ms)
RefineNet [Lin et al., 2017a]	ResNet-152	-	-	47.3
UNet++ [Zhou et al., 2018]	ResNet-101	-	47.7	-
PSPNet [Zhao et al., 2017a]	Dilated-ResNet-101	157.0	47.8	-
Ding *et al.* [Ding et al., 2018]	ResNet-101	-	51.6	-
EncNet [Zhang et al., 2018]	Dilated-ResNet-101	192.1	52.6	-
HRNet [Sun et al., 2019]	HRNetV2-W48	82.7	54.0	48.3
NRD [Zhang et al., 2021a]	ResNet-101	42.9	54.1	49.0
GFFNet [Li et al., 2020c]	Dilated-ResNet-101	-	54.3	-
EfficientFCN [Liu et al., 2020]	ResNet-101	52.8	55.3	-
OCRNet [Yuan et al., 2020]	HRNetV2-W48	143.9	56.2	-
SETR-MLA [Zheng et al., 2021a]	ViT-Large	318.5	-	55.8
Segmenter [Strudel et al., 2021]	ViT-Large	346.2	-	59.0
SenFormer [Bousselham et al., 2021]	Swin-Large	-	64.0	-
SegViT (*Shurnk*, ours)	ViT-Large	186.9	62.3	57.4
SegViT (ours)	ViT-Large	321.6	65.3	59.3
SegViT (*Shurnk*, ours)	BEiTv2-Large	**179.3**	64.91	59.92
SegViT (ours)	BEiTv2-Large	329.7	**67.14**	**61.63**

too much performance. Our method achieves State-of-the-art 58.2% (MS) in terms of mIoU with the BEiTv2 Large backbone. To ensure a fair comparison, we evaluate our SegViT module with the BEiT-v2 large backbone on a crop size of 512×512, which consumes 374.0 GFlOPs. Our approach achieves a slightly better performance of 56.5% mIoU compared to Mask2former-Swin-L, which achieves 56.1% with 402.7 GFlops on a crop size of 640 × 640. Additionally, our *Shrunk* version with a computational cost reduction of around 50% (308.8 GFlops) achieves a competitive performance of 57.0%



(a) Segmenter ViT-Large(51.8) | (b) StructToken ViT-Large(52.8) | (c) UperNet BEiT-Large(56.3) | **(d) SegViT V2** BEiT-2Large(58.0)

FIGURE 4.7. Visuals results of different segmentation networks and plain ViT backbones on the ADE20K validation set[Zhou et al., 2017a]. It includes the following models: (a) Segmenter [Strudel et al., 2021] with ViT large, (b) StructToken [Lin et al., 2022] with ViT large, (c) UPerNet [Xiao et al., 2018] with BEiT large, and (d) SegViT V2 with BEiTv2 large. The results demonstrate that our methods effectively generate accurate segmentation masks and unlock the potential of plain ViT. Zoom in for a better view.

(MS) in terms of mIoU. Figure. 4.7 shows the visual results of different segmentation methods. In contrast to other methods that often confuse similar classes and

misclassify related concepts, our SegViT stands out by more precise object boundary delineation and achieving accurate segmentation of complete objects, even in cluttered scenes.

Results on COCO-Stuff-10K.

Table. 4.2 shows the result on the COCO-Stuff-10K dataset. Our method achieves 50.3% which is higher than the previous state-to-the-art StrucToken by 1.2% with less computational cost. Our *Shrunk* version achieves 49.4% with 224.8 GFLOPs, which is similar to the computational cost of a dilated ResNet-101 backbone but with much higher performance.

Results on PASCAL-Context.

Table. 4.3 shows the results on the PASCAL-Context dataset. We follow HRNet [Sun et al., 2019] to evaluate our method and report the results under 59 classes (without background) and 60 classes (with background). Using full SegViT structure without adopting *Shrunk*, we reach mIoU of 67.14% and 61.63% respectively for those two metrics, outperforming the state-of-the-art methods using the ViT backbones with less computational cost. By applying *Shrunk* architecture, the computational cost in terms of GLOPs is reduced by 42% and 45%, respectively. SegViT with *Shrunk* achieves the best trade-off between accuracy and speed among all methods on the PASCAL-Context dataset.

4.3.4 Ablation Study

In this section, we conduct extensive ablation studies to show the effectiveness of our proposed methods.

Effect of the ATM module.

We conducted an analysis to evaluate the impact of using the proposed ATM module as an encoder. The results are summarized in Table. 4.4. To establish a baseline for comparison, we introduced SETR-naive, which utilizes two 1×1 convolutions to directly derive per-pixel classifications from the final layer of the ViT-Base transformer output. From the results, it is evident that applying the ATM module under the supervision of a conventional cross-entropy loss leads to a performance improvement of 0.5%. However, the performance gains become much more substantial when we decouple the classification and mask prediction processes, supervising each separately. This approach results in a significant performance boost of 3.1%, highlighting the efficacy of the ATM module in enhancing semantic segmentation performance.

TABLE 4.4. Comparisons between our proposed ATM module with SETR [Zheng et al., 2021a]. 'CE loss' indicates the cross-entropy loss commonly used in semantic segmentation. The experiments on the ADE20k dataset are carried out using the ViT-Base backbone.

Decoder	Loss	mIoU (ss)
SETR	CE loss	46.5
ATM	CE loss	47.0 (+0.5)
ATM	$\mathcal{L}_{mask}$ loss	**49.6 (+3.1)**

Ablation of the feature levels.

The effects of using multiple-layer inputs from the backbone to the ATM modules are presented in Table. 4.5. The incorporation of feature maps from lower layers leads to a notable performance improvement of 1.3%. We further investigated the impact of including more layers of features and observed additional gains in performance. After empirical testing, we determined that utilizing three layers yielded optimal results, resulting in an overall mIoU boost of 1.7%. These ablation studies confirm the effectiveness of our proposed ATM decoder and highlight the advantage of incorporating multi-layer features into the segmentation structure. This integration significantly enhances the performance of semantic segmentation tasks.

TABLE 4.5. Ablation results of using different layer inputs to the SegViT structure on ADE20K dataset using ViT-Base as the backbone. Involving multi-layer features can bring obvious performance gains.

	Used layers	mIoU (ss)
Single	[12]	49.6
Cascade	[6, 12]	50.9 (+1.3)
Cascade	[6, 8, 12]	**51.3 (+1.7)**
Cascade	[3, 6, 9, 12]	51.2 (+1.6)

SegViT on hierarchical base models.

We conducted an analysis to evaluate the performance of SegViT on hierarchical base models. For comparison, we selected two competitive methods, Maskformer [Cheng et al., 2021b] and Mask2former [Cheng et al., 2022a]. The results are presented in Table. 4.6 indicate that, even though our method was not specifically designed for hierarchical base models, we are still able to achieve competitive performance while maintaining computational efficiency. This demonstrates the applicability of our SegViT approach to various types of ViT-Base models.

TABLE 4.6. The experiments use the Swin-Tiny [Liu et al., 2021] backbone and are carried out on the ADE20K dataset. The GFLOPs are measured at single-scale inference with a crop size of 512 × 512.

Method	mIoU (ss)	GFLOPs
Maskformer [Cheng et al., 2021b]	46.7	57.3
Mask2former [Cheng et al., 2022a]	**47.7**	73.7
SegViT (Ours)	47.1	**48.0**

Ablation of *Shrunk* strategies.

In this section, we conduct an ablation study to assess the effectiveness of different SegViT structures. Table. 4.7 presents the impact of various techniques employed in each SegViT structure, including query upsampling (QU), query downsampling (QD), Edged query downsampling (EQD) techniques, and segmentation heads. When replacing the SETR head with the ATM head in the 'Plain' structure, a notable performance improvement of 3.1% is observed. This highlights the effectiveness of

TABLE 4.7. Ablation results of *Shrunk* version on the ADE20K dataset. We explored various shrink strategies. The GFLOPs are measured at single-scale inference with a crop size of 512×512 on the ViT-Base backbone. QD: query-based downsampling. QU: query-based upsampling. QD_{layer} indicates which layer to apply the QD. EQD: Edged query downsampling. QD_{method} indicates the downsampling method for QD.

Structure	QD	QU	QD_{layer}	QD_{method}	Head	mIoU (ss)	GFLOPs
Plain	-	-	-	-	SETR	46.5	107.3
Plain	-	-	-	-	ATM	49.6 (+3.1)	115.8
Plain	✓	-	6	2x2	ATM	46.9 (+0.4)	74.1
Shrunk	✓	✓	6	2x2	ATM	**50.0 (+3.5)**	97.1
Shrunk	✓	✓	0	2x2	ATM	43.3(-3.2)	46.1
Shrunk	✓	✓	0	2x2-EQD	ATM	**49.9(+3.4)**	**74.6**

TABLE 4.8. Ablation results of different decoder methods with their corresponding feature merge types and loss types. ViT-Base is employed as the backbone for all the variants.

	Multi-level Features		Loss Types			
Decoder	FPN	Token Merge	Pixel level	Dot product	Attention Mask	mIoU (ss)
SETR-MLA [Zheng et al., 2021a]	✓		✓			48.2
Segmenter [Strudel et al., 2021]			✓			49.0
MaskFormer [Cheng et al., 2021b]	✓			✓		46.7
Ours-Variant 1					✓	49.6
Ours-Variant 2		✓		✓		50.6
Ours		✓			✓	51.2

TABLE 4.9. Ablation of the QD module in terms of the targets and methods to down-sample. The experiments are carried out on the ViT-Large backbone of ADE20K dataset.

Applied to	Methods	mIoU (ss)
Q	Conv	44.5
Q, K, V	Nearest	52.6
Q	Nearest	**53.9**

TABLE 4.10. Comparisons for various ViT pre-training schedules on the validation set of ADE20K. All results are reported in single-scale inference. The default configuration for these base models is pre-trained on ImageNet-1K with 224 * 224 resolutions. '*' means the models use the backbone weights pre-trained with 384 * 384 resolutions. '†' means the base models pre-trained on imagenet-21K. The proposed SegVit head has a less computational cost and performs better than UPerNet among all pre-training variants.

Backbone	SegViT mIoU	Head FLOPs	UPerNet mIoU	Head FLOPs	ImageNet Acc
MAE Base [He et al., 2022]	49.22 (▲1.12)	6.89(▼329.73)	48.1	336.62	83.66
CLIP Base [Radford et al., 2021]	50.76 (▲1.16)	6.89(▼329.73)	49.6	336.62	80.20
CAE Base [Chen et al., 2022]	50.42 (▲0.22)	6.89(▼329.73)	50.2	336.62	83.90
iBot Base [Zhou et al., 2022]	50.58 (▲0.58)	6.89(▼329.73)	50.0	336.62	84.00
Augreg Base*† [Steiner et al., 2021]	51.30 (▲2.66)	6.89(▼329.73)	48.6	336.62	85.49
DEiT v3 Base† [Touvron et al., 2022]	52.40 (▲0.60)	6.89(▼329.73)	51.8	336.62	85.70
BEiT v2 Base† [Peng et al., 2022]	53.97 (▲0.47)	6.89(▼329.73)	53.5	336.62	86.50
Augreg Large*† [Steiner et al., 2021]	54.60 (▲2.50)	16.36(▼1,366.33)	52.1	1382.69	85.59
DEiT v3 Large*† [Touvron et al., 2022]	55.81 (▲1.21)	16.36(▼1,366.33)	54.6	1382.69	87.70
BEiT v2 Large [Touvron et al., 2022]	58.00 (▲1.3)	16.36(▼868.28)	56.7	884.64	87.30

the ATM head in enhancing the performance of the baseline structure. However, introducing QD to the 'Plain' structure with the ATM head leads to a performance drop of 2.7%, indicating information loss during the downsampling process. To address this, we utilize QU to create the 'Shrunk' structure, which retains performance. Yet, applying QD at the 6th layer still incurs significant computational cost. To reduce this overhead, we move the QD application to before the initial transformer layer, resulting in a substantial decrease in computational cost but sacrificing performance (-6.7%). Nevertheless, by incorporating EQD, we successfully recover the lost performance while maintaining a low computational cost. This validates the efficacy of EQD in overcoming information loss and striking a balance between performance and efficiency in the SegViT structures.

Ablation studies on decoder variances.

Different decoder methods are associated with specific feature merge types and loss types. In Table 4.8, we compare the designs of various decoders on a plain ViT backbone. For hierarchical base models like Swin, the resolution of the feature maps in each stage is reduced. Consequently, the adoption of the Feature Pyramid Network (FPN) is necessary to obtain feature maps with larger resolutions and rich semantic information. However, in Table 4.8, we observe that the FPN structure does not perform well with plain vision transformers. In the case of plain ViT base models, the resolution is maintained, and the feature map of the last layer contains the most comprehensive semantic information. Hence, our proposed method, which utilizes tokens to merge features from different levels, achieves better performance. By simply replacing the FPN structure with the ATM-based token merge, we improve the performance from 46.7% to 50.6%. Regarding the loss type, the pixel-level loss refers to the conventional cross-entropy loss applied to the feature map. The dot product loss corresponds to the loss utilized in [Carion et al., 2020b] and [Cheng et al., 2021b]. Attention mask loss indicates that mask supervision is directly applied to the similarity map generated by the ATM during attention calculation. By adding loss supervision on the attention mask, as in our proposed method, the performance improves by 0.6%.

Ablation for the QD module.

The motivation behind using QD is to leverage the pre-trained weights of the backbone. As shown in Table 4.9, if we employ a stride-2 convolution with learnable

parameters to downsample the query, it will disrupt the pre-trained weights and result in a significant performance decrease. Applying down-sampling to both the query and the key-value pairs would inevitably lead to information loss during the down-sampling process, which is evident in the weaker performance observed. Through our investigations, we have found that applying 2×2 nearest down-sampling exclusively to the query in the QD module yields better results. This approach allows us to preserve the pre-trained weights of the backbone while achieving the desired down-sampling effect.

4.3.5 Application 1: A Better Indicator for Feature Representation Learning

Background.

Semantic segmentation serves as a fundamental vision task that has been extensively employed in previous research to assess the representation learning capabilities of weakly, fully, and self-supervised base models [Chen et al., 2022; He et al., 2022; Peng et al., 2022; Touvron et al., 2022]. In prior work, the UPerNet decoder structure has been commonly used for semantic segmentation. However, the UPerNet decoder may not be a suitable indicator for evaluating the feature representation ability of the base model. This is primarily due to its heavier computational requirements and slower convergence rate. Additionally, the feature representation obtained by the base model can vary significantly due to different training strategies employed during the fine-tuning process on semantic segmentation datasets. Consequently, the task of semantic segmentation may not effectively evaluate the feature representation ability of pre-trained models.

Experiment settings.

This section presents a comprehensive evaluation of our proposed SegVit on various weakly/fully/self-supervised vision transformers, including those proposed by He et al. [He et al., 2022], Chen et al. [Chen et al., 2022], Touvron et al. [Touvron et al., 2022], and the BEiT model [Peng et al., 2022]. We demonstrate that our method outperforms UPerNet [Xiao et al., 2018] in both self-supervised and multiple modality base models, achieving state-of-the-art performance. Notably, our approach achieves superior performance to UPerNet while utilizing only 5% of the computational cost in terms of the decoder head. Table. 4.10 illustrates that our proposed SegViT head consistently outperforms UPerNet on all base models. For the ViT-Base, our method improves the performance of UPerNet on the CLIP model by 1.16% while significantly reducing the computational cost. Similar observations can be made for ViT-Large base models. Furthermore, compared to UPerNet, our proposed SegViT head exhibits a better alignment between the growth trend of segmentation accuracy and the classification accuracy on ImageNet. This clearly demonstrates the superior efficiency of our SegViT head compared to UPerNet, making it a more suitable indicator for feature representation learning in base models.

4.3.6 Application2: Continual Semantic Segmentation

Due to the decoupling of class prediction and mask segmentation in our proposed SegVit decoder, we are inherently suitable for a continuous learning setting. This characteristic allows us to learn new classes by solely fine-tuning the class proxy (the class token), leveraging the powerful representation ability of the plain vision

TABLE 4.11. CSS results on ADE20k in mIoU (%) on 100-50 and 100-10 settings. The relative mIoU reduction (▼) compared with the joint training for each method is reported.

Method	**100-50** (2 tasks)				**100-10** (6 tasks)			
	0-100	101-150	*all*	*avg*	0-100	101-150	*all*	*avg*
ILT [Michieli and Zanuttigh, 2019]	18.29 (▼26.1)	14.40 (▼13.8)	17.00 (▼22.0)	29.42	0.11 (▼44.2)	3.06 (▼25.1)	1.09 (▼37.9)	12.56
MiB [Cermelli et al., 2020]	40.52 (▼3.9)	17.17 (▼11.0)	32.79 (▼6.2)	37.31	38.21 (▼6.1)	11.12 (▼17.1)	29.24 (▼9.8)	35.12
SDR [Michieli and Zanuttigh, 2021]	40.52 (▼3.8)	17.17 (▼11.0)	32.79 (▼6.2)	37.31	37.26 (▼7.1)	12.13 (▼16.1)	28.94 (▼10.1)	34.48
PLOP [Douillard et al., 2021]	41.76 (▼2.6)	14.52 (▼13.7)	32.74 (▼6.3)	37.73	38.59 (▼5.8)	14.21 (▼14.0)	30.52 (▼8.5)	34.48
REMINDER [Phan et al., 2022]	41.55 (▼2.8)	19.16 (▼9.0)	34.14 (▼4.9)	38.43	38.96 (▼5.4)	21.28 (▼6.9)	33.11 (▼5.9)	37.47
RCIL [Zhang et al., 2022b]	42.35 (▼2.0)	18.47 (▼9.7)	34.45 (▼4.6)	38.48	29.42 (▼15.0)	13.49 (▼14.0)	28.36 (▼10.0)	29.93
Oracle - ResNet backbone	44.34	28.21	39.00	-	44.34	28.21	39.00	-
MiB [Cermelli et al., 2020]	43.43 (▼6.65)	30.63 (▼12.24)	39.19 (▼8.32)	38.66	39.15 (▼16.04)	20.37 (▼41.63)	34.17 (▼20.07)	39.53
PLOP [Douillard et al., 2021]	43.82 (▼6.03)	26.23 (▼24.84)	37.99 (▼11.13)	38.06	43.25 (▼7.25)	24.13 (▼30.86)	36.25 (▼15.21)	40.28
REMINDER [Phan et al., 2022]	44.66 (▼4.22)	26.76 (▼23.33)	38.73 (▼9.40)	38.43	43.28 (▼7.18)	24.33 (▼30.29)	37.10 (▼13.22)	41.76
Oracle - ViT backbone	46.63	34.90	42.75	-	46.63	34.90	42.75	-
SegViT-CL (ours)	**53.64 (▼0.5)**	**40.00 (▼5.6)**	**49.09 (▼2.2)**	46.82	**53.77 (▼0.3)**	**35.54 (▼10.0)**	**47.70 (▼3.6)**	50.59
Oracle	54.11	45.60	51.28	-	54.11	45.60	51.28	-

transformer while keeping the old parameters frozen. To validate the effectiveness of this new approach to continual learning, we conducted quick fine-tuning experiments following previous continuous learning settings.

Experiment settings.

Continual Semantic Segmentation (CSS) has two settings [Cermelli et al., 2020]: disjoint and overlapped. In the disjoint setup, all pixels in the images at each step belong to either the previous classes or the current class. In the overlapped setting, the dataset of each step contains all the images that have pixels of at least one current class, and all pixels from previous and future tasks are labeled as background. The overlapped setting is more realistic and challenging, thus we evaluate the performance of the overlapped setup on the ADE20k dataset.

Following prior published works [Cermelli et al., 2020; Douillard et al., 2021; Phan et al., 2022], we perform three experiments: adding 50 classes after training with 100 classes (100-50 setting with 2 steps), adding 50 classes each time after training with 50 classes (50-50 setting with 3 steps), adding 10 classes each time sequentially after training with 100 classes (100-10 setting with 6 steps).

Baselines

We conducted a comprehensive comparison of our proposed method against state-of-the-art Continual Semantic Segmentation (CSS) techniques, including RCIL [Zhang et al., 2022b], PLOP [Douillard et al., 2021], REMINDER [Phan et al., 2022], SDR [Michieli and Zanuttigh, 2021], and MiB [Cermelli et al., 2020]. To ensure fair comparisons, existing methods were evaluated using DeepLabV3 [Chen et al., 2017c] with ResNet101 and ViT-Base backbones that were pre-trained on ImageNet-21k. The reported results for PLOP, RCIL, and REMINDER were obtained based on the codebases provided by the respective authors. Furthermore, we included the performance of the Oracle model, which represents the upper bound achieved by jointly training on all available data, serving as a benchmark for each method.

Metrics. We evaluate the model performance by five mIoU metrics. First, we compute mIoU for the base classes $\mathcal{C}^0$, which reflects model rigidity: the model's resilience to catastrophic forgetting. Second, we compute mIoU for all incremented classes $\mathcal{C}^{1:T}$, which measures plasticity: the model capacity in learning new tasks. Third, we compute the mIoU of all classes in $\mathcal{C}^{0:T}$ (*all*), which shows the overall performance of models. Fourth, we report the average of mIoU (*avg*) measured step after step as proposed by [Douillard et al., 2021], which evaluates performance over the

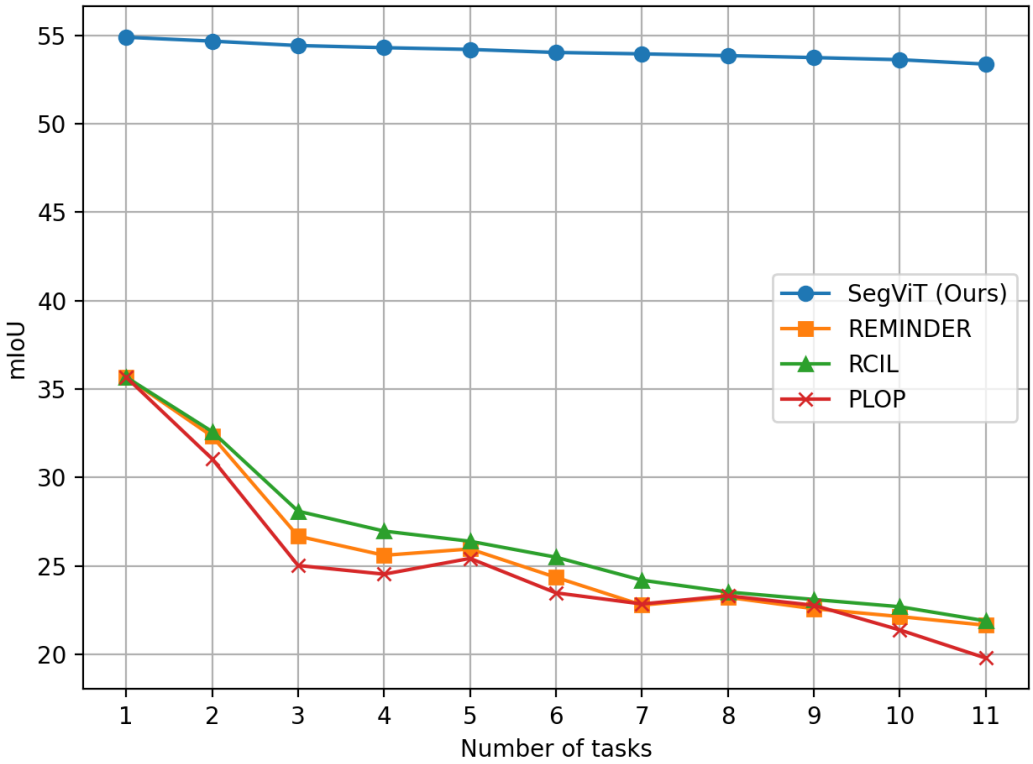


FIGURE 4.8. mIoU of recent CSS methods on the first 100 base classes after incrementally learning new tasks on 100-5 settings with 11 tasks.

TABLE 4.12. Performance drop (degree of forgetting) of all classes grouped by tasks in the 100-10 setting. We report the class mIoU when the model first learns the task, and the mIoU when the model last learns it.

Tasks	101-110	111-120	121-130	131-140	141-150	avg
First Time	34.93	39.78	41.10	36.22	27.95	35.99
Last Time	34.51	39.30	40.86	35.09	27.95	35.54
Forgetting	▼0.42	▼0.48	▼0.24	▼1.12	▼0	▼0.45

entire continual learning process. To ensure fair comparisons, we evaluate the relative performance of each CSS method in terms of relative mIoU reduction compared with its Oracle model jointly trained on all data.

Results and Discussion.

Table 4.11 shows the results of different CSS methods on ADE20k. Our SegViT-CL consistently outperforms existing methods in *all* mIoU on both settings. In terms of mIoU reduction, the proposed SegViT-CL only decreases the mIoU of the Oracle model by 2.2% on the 100-50 setting, which is two times better than the second-best method, RCIL with ResNet backbone with 4.6% reduction. This substantial enhancement over existing methods underlines the effectiveness of our proposed method in the continual semantic segmentation paradigm. On a long CL setting 100-10 with 6 tasks, ours is almost forgetting-free with a marginal mIoU reduction of 0.3%, while recent CSS methods significantly suffer from forgetting with at least 5.4% mIoU reduction. Using the ViT backbone, existing methods including MiB, REMINDER and

PLOP suffer from higher mIoU reductions than using the ResNet backbone. Compared with the Oracle, MiB [Cermelli et al., 2020], PLOP [Douillard et al., 2021], and REMINDER [Phan et al., 2022] decrease the mIoU by 20.07%, 15.21% and 13.22% respectively in the 100-10 setting. The current CSS methods are less effective for ViT architecture. This highlights the need for developing a specialized ViT architecture that is robust to forgetting.

To evaluate the forgetting of every task in the 100-10 setting, we compute the performance drop at the last step compared with its initial mIoU when the model first learns the task. For example, the initial mIoU of task 2 is the mIoU of class 101-110 evaluated at step 2. Similarly, that of task 3 is the mIoU of class 111-120 reported at step 3. Table. 4.12 shows the performance drop at the last step compared with the initial mIoU of each task. Averaged across 5 tasks, the mIou only drops by 0.45%, which shows that SegViT is robust to forgetting across all tasks in the 100-10 setting. Table. 4.8 shows the mIoU on the base classes after incremental training on many tasks in 100-5, which is a long continual learning setting with 11 tasks. Overall, our SegViT achieves nearly zero forgetting for almost all tasks at the last step. In contrast to previous CSS methods which require partial fine-tuning, the proposed SegViT supports completely freezing old parameters to eliminate interference with past knowledge.

4.4 Conclusion

This chapter presents SegViT, a novel approach for semantic segmentation using plain ViT transformer base models. The proposed method introduces a lightweight decoder head that incorporates the Attention-to-mask (ATM) module. Additionally, a *Shrunk* structure is proposed to reduce the computational cost of the ViT encoder by 50% while maintaining competitive segmentation accuracy. Moreover, this work extends the SegViT framework to address the challenge of continual semantic segmentation, aiming to achieve nearly zero forgetting. By protecting the parameters of old tasks, SegViT effectively mitigates the impact of catastrophic forgetting. Extensive experimental evaluations conducted on various benchmarks demonstrate the superiority of SegViT over UPerNet, while significantly reducing computational costs. The introduced decoder head provides a robust and cost-effective solution for future research in the field of ViT-based semantic segmentation.

Statement of Authorship

Title of Paper	Dynamic token pruning in plain vision transformers for semantic segmentation
Publication Status	☐ Published ☑ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style
Publication Details	Accepted in ICCV 2023

Principal Author

Name of Principal Author (Candidate)	Bowen Zhang
Contribution to the Paper	Proposed the ideas, conducted the experiments and wrote part of the paper
Overall percentage (%)	70
Certification:	This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper.
Signature	Date 10/8/2023

Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);
ii. permission is granted for the candidate in include the publication in the thesis; and
iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

Name of Co-Author	Quan Tang
Contribution to the Paper	Discussion, write the paper and the revision
Signature	Date 10/08/2023

Name of Co-Author	Jiajun Liu
Contribution to the Paper	Discussion and write the revision
Signature	Date 1/9/2023

Name of Co-Author	Fagui Liu
Contribution to the Paper	Discussion and write the revision
Signature	Date 10/08/2023

Name of Co-Author	Yifan Liu
Contribution to the Paper	Discussion and write the revision
Signature	Date 29/08/2023

Chapter 5

Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation

5.1 Introduction

The Transformer [Vaswani et al., 2017b] is a remarkable invention because of its exceptional capability to model long-range dependencies in natural language processing. It has been extended to computer vision applications and is known as the Vision Transformer (ViT), by treating every image patch as a token [Dosovitskiy et al., 2021a]. Benefiting from the global multi-head self-attention, competitive results have been achieved on various vision tasks, *e.g.* image classification [Dosovitskiy et al., 2021a; Yuan et al., 2021], object detection [Carion et al., 2020a; Zhu et al., 2021] and semantic segmentation [Cheng et al., 2021a, 2022b; Zhang et al., 2022a]. However, heavy computational overhead still impedes its broad application, especially in resource-constrained environments. In semantic segmentation, the situation deteriorates since high-resolution images generate numerous input tokens. Therefore, redesigning lightweight architectures or reducing computational costs for ViT has attracted much research attention.

Since the computational complexity of vision transformers is quadratic to the token number, decreasing its magnitude is a direct path to lessen the burden of computation. There has been a line of works studying persuasive techniques of token pruning regarding the image classification task. For example, DynamicViT [Rao et al., 2021] determines kept tokens using predicted probability by extra subnetworks, and EViT [Liang et al., 2022b] reorganizes inattentive tokens by computing their relevance with the [*cls*] token. Nevertheless, *removing tokens, even if they are inattentive, can not be directly extended to semantic segmentation since a dense prediction is required for every image patch.* Most recently, Liang *et al.* [Liang et al., 2022a] proposed a token reconstruction layer that rebuilds clustered tokens to address the issue.

In this work, a fresh angle is taken and breaks out of the cycle of token clustering or reconstruction. Motivated by humans' coarse-to-fine and easy-to-hard segmentation process, we progressive grade tokens by their difficulty levels at each stage. Hence for easy tokens, their predictions can be finalized in very early layers and their forward propagation can be halted early on. Consequently, only hard tokens are processed in the following layers. We refer to the process as the *early exit* of tokens. Figure 5.1 gives an illustration. The main body of the relatively larger objects in the image is first recognized and their process is ceased, while deeper layers progressively handle those challenging and confusing boundary regions and smaller objects. These predictions from the staged, early-exiting process can be used together with those from the

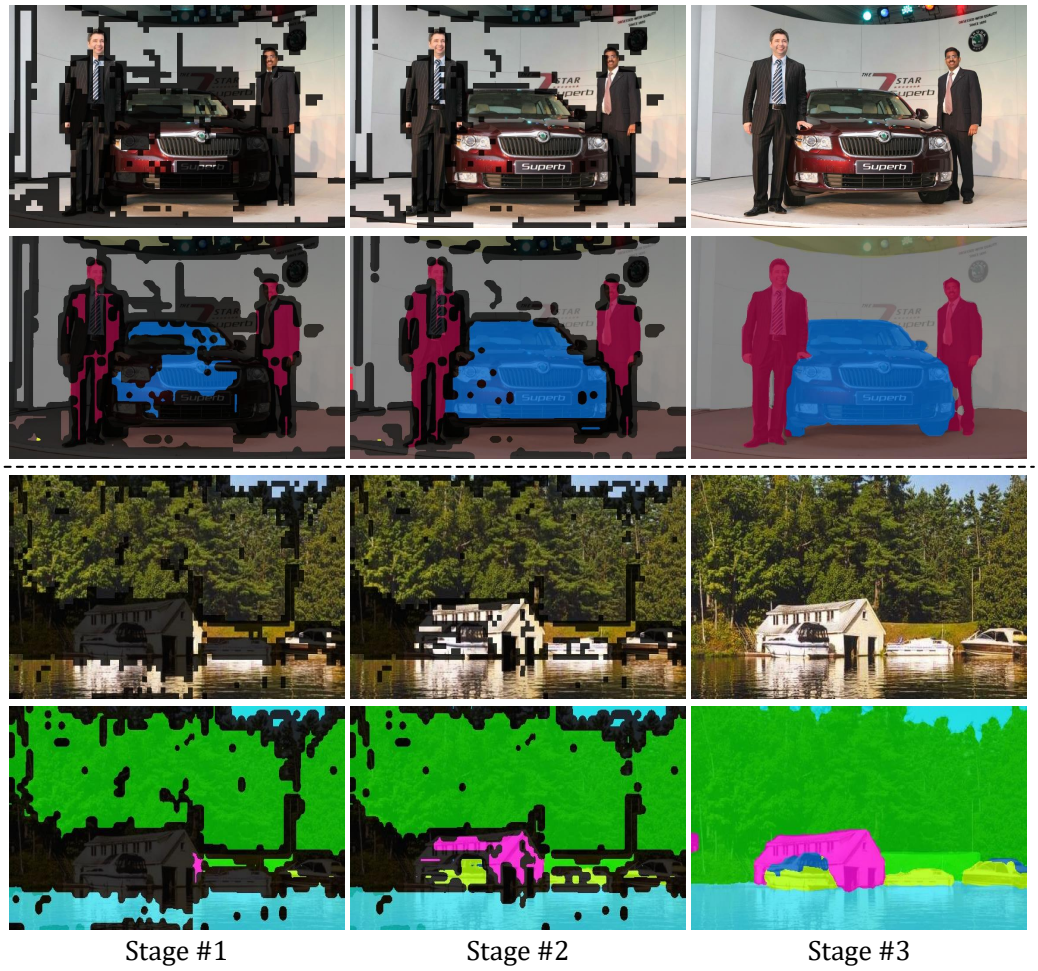


FIGURE 5.1. **Illustration of token difficulty levels by three stages using ADE20K dataset**. The network is naturally split into stages using inherent auxiliary blocks. Each sextuplet presents the early-exited/pruned tokens and their corresponding predictions successively for an image, where bright areas represent early-exited easy tokens at the current stage, while the dark ones are the kept hard tokens for the following computing.

completed inference. Since both outputs then form the final results jointly, it requires no token reconstruction operation and results in a simple yet effective form of efficient ViT for segmentation.

This work introduces a novel Dynamic Token Pruning (DToP) paradigm in plain vision transformers for semantic segmentation. Given that auxiliary losses [Zhao et al., 2017b; Zheng et al., 2021b] are widely adopted, DToP divides a transformer into stages using the inherent auxiliary blocks without introducing extra modules and calculations. *While previous works discard auxiliary predictions irrespectively, we make good use of them to grade all tokens' difficulty levels.* The intuition of such design lies in the dissimilar recognition difficulties of image patches represented by individual tokens. Easy tokens are halted and pruned early on in the ViT, while hard ones are kept to be computed in the following layers. We note that having this observation and shifting from auxiliary-loss-based architecture to DToP for token reduction is a non-trivial contribution. A possible situation exists where objects consisting of only extremely easy tokens, *e.g.* sky. As a result, DToP completely discards tokens from

easy-to-recognize categories in early layers, and this causes a severe loss of contextual information for the few remaining tokens in their computations. To fully utilize the inter-class feature dependencies and uphold representative context information, we keep k highest confidence tokens for each semantic category during each pruning process.

Contributions are summarized as follows:

- We introduce a dynamic token pruning paradigm based on the early exit of easy-to-recognize tokens for semantic segmentation transformers. The finalized easy tokens at intermediate layers are pruned from the rest of the computation, and others are kept for continued processing.
- We uphold the context information by retaining k highest confidence tokens for each semantic category for the following computation, which improves the segmentation performance by guaranteeing that enough contextual information is available even in extremely easy cases.
- We apply DToP to mainstream semantic segmentation transformers and conduct extensive experiments on three challenging benchmarks. Results suggest that DToP can reduce up to 35% computation costs without a notable accuracy drop.

5.2 Our Methods

This work introduces a Dynamic Token Pruning method based on the early exit of tokens, which expedites plain vision transformers for semantic segmentation. We detail the paradigm in this section.

5.2.1 Preliminary

A conventional vision transformer [Dosovitskiy et al., 2021a] splits an image $X \in \mathbb{R}^{3\times H\times W}$ into different patches. We then obtain a sequence of $\frac{HW}{P^2} \times C$ via patch embedding. H and W represent the image resolution, P is the patch size and C is the feature dimension. Let $N = \frac{HW}{P^2}$ be the length of the input sequence, *i.e.* the number of tokens. Vision transformers are position-agnostic, and we generally add positional encoding to represent the spatial information of each token. The resulting sequence is denoted as $Z_0 \in \mathbb{R}^{N\times C}$, which serves as the input.

Vision transformers are usually developed from repeated units that contain a multi-head self-attention (MHSA) module and a feed-forward network (FFN). Layer normalization (LN) [Ba et al., 2016] and residual connection [He et al., 2016a] are employed within such units. We refer to a unit as one layer indexed by $l \in \{1, 2, ..., L\}$, and the output of each layer is marked as Z_l.

$$\begin{aligned} Z'_l &= \texttt{MHSA}(\texttt{LN}(Z_{l-1})) + Z_{l-1}, \\ Z_l &= \texttt{FFN}(\texttt{LN}(Z'_l)) + Z'_l. \end{aligned} \tag{5.1}$$

Note that FFN includes a non-linear activation function, *e.g.* GeLU [Hendrycks and Gimpel, 2016].

5.2.2 Dynamic Token Pruning

Since a token is a natural representation of an image patch, we can finalize the prediction for easy tokens in advance without the need for complete forward computing by mimicking the segmentation process of humans. We refer to it as the *early exit*

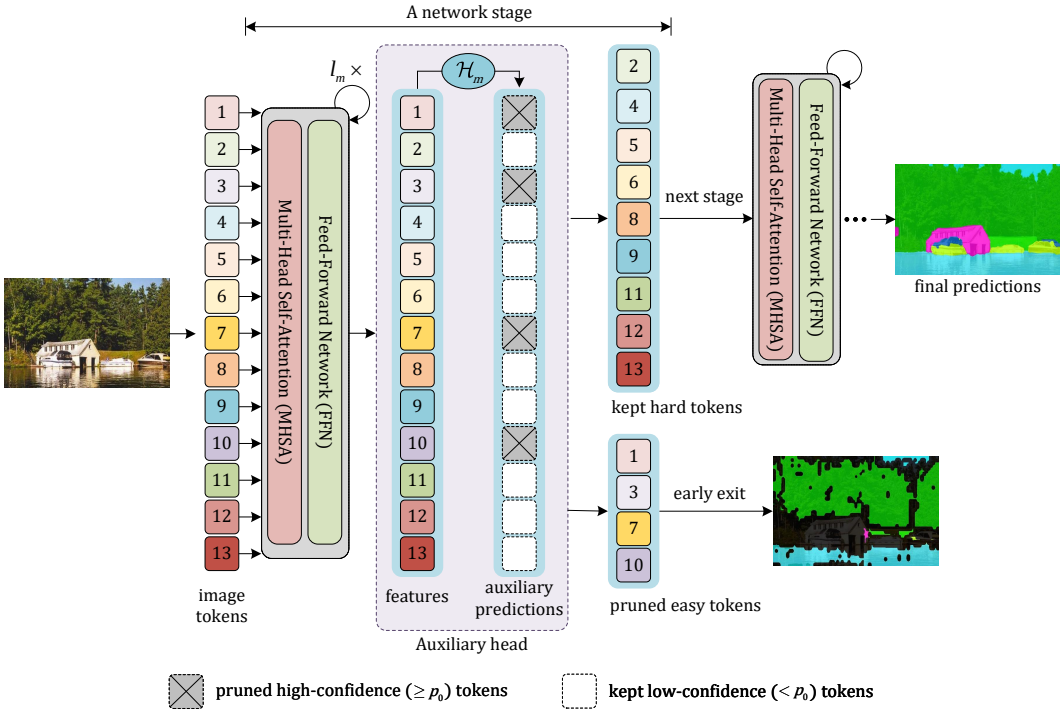


FIGURE 5.2. **Illustration of the proposed DToP framework.** Given an existing plain vision transformer, we divide it into stages using the inherent auxiliary heads. At the final layer (indexed by l_m) of the m-th stage, we use the auxiliary block $\mathcal{H}_m$ to grade all token difficulty levels. We finalize the predictions of high-confidence easy tokens at the current stage and handle other low-confidence hard tokens in the following stages. The retained k highest confidence tokens for each semantic category to uphold representative context information is not presented for simplicity. Predictions from each stage jointly form the final results.

of tokens, where easy tokens are halted and pruned in the early stages while hard ones are preserved for calculation at the latter stages. By doing so, fewer tokens are processed in the following layers, significantly reducing the computation costs.

As shown in Figure 5.2, we divide a plain vision transformer backbone into M stages using its inherent auxiliary blocks $\mathcal{H}_m$ ($m \in \{1, 2, ..., M\}$) at the end of each stage. Let $P_m \in \mathbb{R}^{N \times K}$ represent the predicted results at the m-th stage, where K is the number of semantic categories. Suppose that tokens have finished l_m layers of forward propagation at this point, then:

$$P_m = \mathcal{H}_m(Z_{l_m}). \tag{5.2}$$

$p_{m,n}$ coming from P_m is the maximum predicted probability of the n-th token. Previous works adopt P_m to calculate auxiliary losses during training and discard them irrespectively during inference. This work highlights that easy tokens can be correctly classified with high predictive confidence in these auxiliary outputs (*i.e.* P_m). The proposed DToP expects to fully explore their potential ability to tell apart easy and hard tokens during both training and inference.

Inspired by [Hendrycks and Gimpel, 2017], we grade all token difficulty levels using P_m based on a simple criterion. Assume a large confidence threshold p_0, *e.g.* 0.9. Easy tokens are classified with higher than 90% scores, while hard ones are classified with lower scores. Since confident predictions for easy tokens are obtained, we prune them and halt their continued forward propagation. Hard tokens are reserved in computing

in the following layers to achieve a reliable prediction. In other words, we prune the n-th token in Z_{l_m} if $p_{m,n} \geqslant p_0$, otherwise we keep it. After propagating an image through the whole network, we combine the predicted token labels from each stage to form the final results. By default, the pruning process is executed twice, with each iteration incurring a minimal computational cost due to the use of a straightforward classifier at a 1/16 resolution. During inference, the overall computational expenses, including those associated with the pruning process, are considered for comparison against alternative methods.

5.2.3 Query Matching Auxiliary Block

Within the DToP framework, the auxiliary block for grading all token difficulty levels should follow two principles: capable of accurately estimating token difficulty levels and with a lightweight architecture. Therefore, we take the most recent attention-to-mask module (ATM) [Zhang et al., 2022a] to achieve this goal. Specifically, a series of learnable class tokens exchange information with the encoder features using a transformer decoder. The output class tokens are used to get class probability predictions. The attention score regarding each class token is used to form a mask group. The dot product between the class probability and group masks produces the final prediction.

Two modifications are made to adapt ATM into the DToP framework. First, we decrease the number of layers in ATM as we observe no significant performance perturbation in the DToP framework with the original setting, which also guarantees a low computational overhead. Second, we decouple multiple cascaded ATM modules and use them as separate auxiliary segmentation heads, each with individual learnable class tokens. We note that we take the powerful ATM module to grade all token difficulty levels as an example, as a reliable estimation of tokens' segmentation difficulty may lead to a good accuracy-computation trade-off. Any other existing segmentation heads are of the same effect (see [Xie et al., 2021; Zhao et al., 2017b; Zheng et al., 2021b] for examples). In Section 5.3, we also provide experiments with the regular FCN head [Long et al., 2015b] to validate the generality of DToP.

5.2.4 Upholding Context Information

Scenarios exist where all tokens of a specific semantic category are extremely easy to recognize, *e.g.* sky. Such tokens may be entirely removed in early layers, resulting in a loss of context information in the following layers of calculation. Practices [Yu et al., 2018c, 2020c] indicate that fully exploiting the inter-category contextual information improves the overall semantic segmentation accuracy. To this end, we keep k highest confidence tokens for each semantic category during each pruning process. *Only the categories that appear in the current image are considered.* Given a specific semantic category, if the number of tokens with a higher than p_0 score is more than k, then the top-k of them are kept. Otherwise, we keep the actual number of them. These category-known tokens join in the calculation along with other low-confidence ones, so semantic information of easy category is preserved for inter-category information exchange, leading to an accurate semantic segmentation.

5.3 Experiments

5.3.1 Datasets and Metrics

ADE20K [Zhou et al., 2017b] is a widely adopted benchmark dataset for semantic segmentation. It contains about $20k$ images for training and $2k$ images for validation. All images are labeled with 150 semantic categories.

COCO-Stuff-10K [Caesar et al., 2018a] dataset contains $9k$ images for training and $1k$ images for testing. Following [Zhang et al., 2022a], we use 171 semantic categories for experiments.

Pascal Context [Mottaghi et al., 2014a] has a total of $10,100$ images, of which $4,996$ images are for training and $5,104$ for validation. It provides pixel-wise labeling for 59 categories, excluding the background.

Following the common convention, we use the mean intersection over union (mIoU) to evaluate the segmentation accuracy and the number of float-point operations (FLOPs) to estimate the model complexity. The computation in DToP is unevenly allocated among easy and hard samples by pruning different numbers of tokens. We thus report the average FLOPs over the entire validation/test dataset.

5.3.2 Implementation Details

We adopt the plain vision transformer incorporating the adapted ATM module as the baseline model, where ATM modules work as auxiliary heads. We follow the standard training settings in *mmsegmentation*[1] and use the same hyperparameters as the original paper. All reported mIoU scores are based on single-scale inputs. k is set to 5 in this work. As changing p_0 within a certain range ($0.90 \sim 0.98$) during training leads to similar results, we empirically fix it to 0.95 for all training processes unless specified.

5.3.3 Ablation Study

We first conduct extensive ablation studies with the ADE20K dataset [Zhou et al., 2017b] using ViT-Base [Dosovitskiy et al., 2021a] as the backbone.

5.3.3.1 Necessity for Model Training

Using auxiliary heads for efficient training is a common convention in the semantic segmentation community, see [Cheng et al., 2022b; Zhao et al., 2017b; Zheng et al., 2021b] for examples. Generally, the auxiliary outputs are discarded at test time. As the proposed DToP grades all token difficulty levels using the auxiliary outputs, we can apply DToP to existing methods off-the-shelf during inference. Therefore, we verify the necessity for model retraining or finetuning under the proposed DToP framework. We denote DToP@Direct as directly applying DToP to the baseline model during inference. DToP@Finetune means finetuning the segmentation heads for $40k$ iterations on the baseline model using DToP, and DToP@Retrain retraining the entire model using DToP for $160k$ iterations.

Results are shown in Table 5.1. We observe that all three settings reduce the computation costs by about 20%, where DToP@Direct and DToP@Retrain lead to a significant accuracy drop while DToP@Finetune performs slightly better. Results suggest that the proposed DToP@Finetune requires only a little extra training time but significantly reduces the computational complexity while maintaining accuracy.

[1] https://github.com/open-mmlab/mmsegmentation

Method	GFLOPs	mIoU(%)
Baseline	109.9	49.7
+ DToP@Direct (♣.0)	87.5	47.9 (-1.8)
+ DToP@Finetune (♣2.5)	86.8	**49.8** (+0.1)
+ DToP@Retrain (♣12.0)	87.5	49.1 (-0.6)

TABLE 5.1. **Comparison of training schemes.** With a short finetuning scheme, the pruned model achieves even better results than the baseline. ♣ means extra training time in hours on 8 NVIDIA A100 cards.

p_0	0.60	0.70	0.80	0.85	0.90	0.95	1.00
GFLOPs	70.2	73.4	77.8	80.7	83.6	86.8	109.9
mIoU(%)	46.8	48.0	49.0	49.3	49.5	**49.8**	49.7

TABLE 5.2. **Ablation for confidence threshold** p_0. The results are evaluated on ADE20K with ATM head.

We adopt the @Finetune setting in the following experiments. Note that the slight fluctuation in FLOPs of the three training schemes comes from varied predictions of auxiliary heads in the individual training processes.

5.3.3.2 Ablation for Confidence Threshold

The confidence threshold p_0 is a crucial hyperparameter that decides the pruned token number in each pruning process and directly affects the trade-off between computation cost and accuracy. Quantitative results are shown in Table 5.2. When $p_0 = 1$, the model degenerates to the baseline architecture. As p_0 decreases, more easy tokens are pruned as well as more unreliable early predictions. We observe that the performance saturates at $p_0 = 0.95$ when using ATM as the segmentation head.

We also verify the value using SETR [Zheng et al., 2021b] (w/ the naive segmentation head described in FCN [Long et al., 2015b]) and show the results in Table. 5.3. We observe that for FCN head $p_0 = 0.98$ may be a better choice. In practice, the value can be chosen empirically with a small validation set. We also note that for SETR, DToP@Direct has already obtained a promising mIoU score of 46.6% that is only 0.4% lower than the baseline but with significantly reduced computation ($\sim$ 23.4%). Some qualitative examples of how the threshold p_0 affects the pruned token number and segmentation accuracy are shown in Figure. 5.3[2].

5.3.3.3 Exploration on Pruning Position

The critical insight of DToP is to finalize the prediction of easy tokens in intermediate layers and prune them in the following calculation by grading all tokens' difficulty levels. Thus the position of auxiliary heads matters. It affects the recognition accuracy of pruned easy tokens and the trade-off between computation cost and segmentation accuracy. We conduct explorations on the pruning position l_m and show the results in Table 5.4. Results demonstrate that dividing the backbone into three stages with

[2]Note that some pruned tokens change their final segmentation due to the attention-to-mask mechanism in ATM but will remain the same in regular FCN heads.

Method	p_0	GFLOPs	mIoU(%)
SETR	-	107.7	47.0
+ DToP@Direct	0.90	74.0	45.6 (-1.4)
+ DToP@Finetune	0.90	72.5	46.3 (-0.7)
+ DToP@Direct	0.95	78.3	46.2 (-0.8)
+ DToP@Finetune	0.95	76.5	46.8 (-0.2)
+ DToP@Direct	0.98	82.5	46.6 (-0.4)
+ DToP@Finetune	0.98	80.6	**47.0** (+0.0)

TABLE 5.3. **Ablation results based on SETR.** About 25% of the tokens can be pruned with no performance dropped.

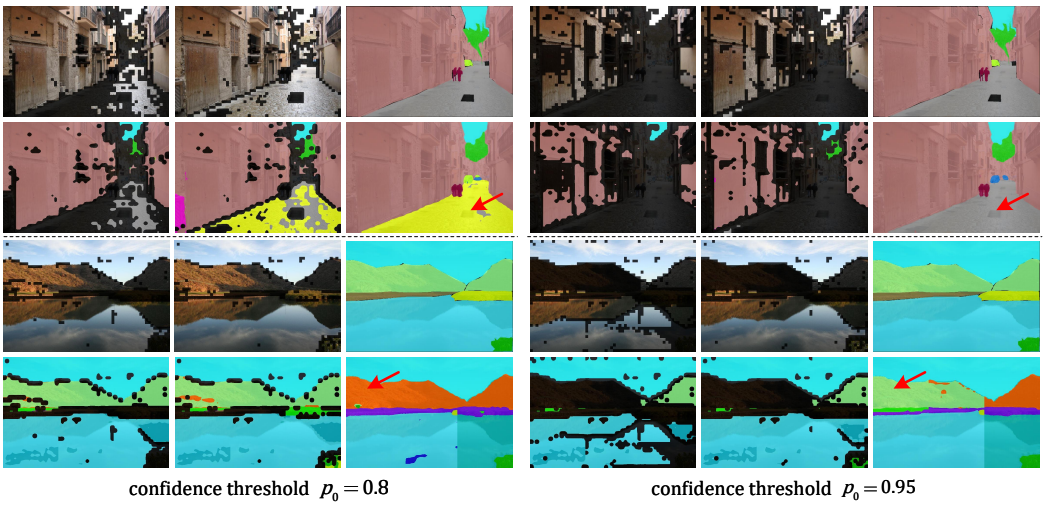


FIGURE 5.3. **Illustration of the effects for different confidence threshold**. Samples are from ADE20K dataset. For each sextuplet, we show the pruned token distribution and the ground truth (first row), as well as its corresponding segmentation results (second row). Bright areas represent pruned tokens, and those in the dark are kept tokens for the following computing. A small p_0 value (left two examples) leads to more pruned tokens in early stages but results in inferior segmentation results (see the red arrow).

token pruning at the 6^{th} and 8^{th} layers achieves an expected trade-off between computation cost and segmentation accuracy. We adopt this setting in all other experiments and note that it may not be optimal on account of limited explorations.

5.3.3.4 Ablation for Pruning Method

After grading all token difficulty levels at the current stage, the specific pruning method is flexible. We experiment with four token pruning methods. Following LC [Li et al., 2017], we remove easy tokens directly without the consideration of halting easy category information. In contrast, this work keeps k highest confidence tokens for each appeared semantic category to uphold representative context information, marked as top-k. An alternative to uphold context information is to average all easy token values into one token for each semantic category. We also prune a fixed proportion of tokens by removing the top 35% highest confidence tokens to evenly allocate computation among images. Results are shown in Table 5.5, where the proposed top-k method

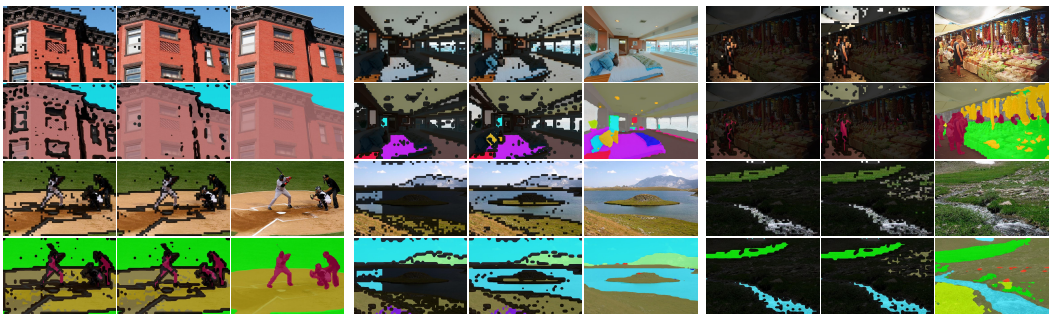


FIGURE 5.4. **Prediction results of three stages during the token pruning processes.** Examples from ADE20K with different image complicity: most tokens are pruned (left group), the majority are pruned (middle group), and very few are pruned (right group). For each sextuplet, we show the pruned token distribution and the corresponding segmentation results at each stage.

Stages	Position	GFLOPs	mIoU (%)
1	Baseline	109.9	49.7
2	{6}	85.7	49.4
2	{8}	92.1	49.4
3	{6, 8}	86.8	**49.8**
4	{3, 6, 8}	74.5	48.3

TABLE 5.4. **Exploration of the pruning position.** The first column indicates the number of divided stages.

outperforms others by a large margin, suggesting its effectiveness. Furthermore, we observe that methods upholding context information, *i.e.* the Average and top-k, are superior to others.

5.3.3.5 Influence of Segmentation Heads

We conduct verification of different segmentation heads in Table 5.6. Results show that the proposed DToP works well in both ATM and FCN settings (first two parts), suggesting the generality. Furthermore, we observe that the mIoU score incurs a sharp drop after applying the proposed DToP in the setting of an inconsistent decoder and auxiliary heads. We assume there is a gap when grading all token difficulty levels using different heads. Adopting ATM as auxiliary heads and FCN as the decoder heads lead to significantly better accuracy than that of the reverse case. We suppose the powerful ATM provides a more accurate estimation of all tokens' difficulty levels and thus obtains superior results.

5.3.4 Application to Existing Methods

We apply the proposed DToP to two mainstream semantic segmentation frameworks in plain vision transformers [Dosovitskiy et al., 2021a]. SETR [Zheng et al., 2021b] uses the naive upsampling decoder, and SegViT [Zhang et al., 2022a] adopts our adapted ATM module. Results are shown in Table. 5.7 using three challenging benchmarks. With an appropriate confidence threshold p_0, the proposed DToP can reduce on average 20% $\sim$ 35% computation cost without notable accuracy degradation. More specifically, SETR with DToP@Finetune reduces 25.2% computation cost (FLOPs

Method	Context	GFLOPs	mIoU(%)
Baseline	-	109.9	49.7
Remove	×	82.6	48.7
Top-35%	×	84.6	48.7
Average	✓	83.5	<u>48.9</u>
Top-k	✓	83.6	**49.5**

TABLE 5.5. **Comparison of different pruning methods.** All models are trained with DToP@Finetune using $p_0 = 0.9$.

Method	Decode	Aux	GFLOPs	mIoU (%)
Baseline	ATM	ATM	109.9	49.7
+ DToP@Finetune	ATM	ATM	83.6	49.5
Baseline	FCN	FCN	107.7	47.0
+ DToP@Finetune	FCN	FCN	80.6	47.0
Baseline	FCN	ATM	107.7	49.6
+ DToP@Finetune	FCN	ATM	83.4	48.4
Baseline	ATM	FCN	109.9	47.9
+ DToP@Finetune	ATM	FCN	73.3	46.9

TABLE 5.6. **Exploration of different segmentation heads.** Results in the second part uses $p_0 = 0.98$ and others 0.9. 'Decode' means the final decoder head and 'Aux' auxiliary head.

107.7G → 80.6G) without mIoU drop on ADE20K and even obtains a slightly better mIoU (58.1% → 58.2%) on Pascal Context dataset. SegViT with DToP@Finetune based on ViT-large reduced about 35% computation with only 0.5% mIoU lower on ADE20K.

A qualitative comparison regarding the pruned token number of different images is presented in Figure. 5.4. We see that most tokens are pruned at very early stages for images of simple scenarios. For complex scene images, most tokens remain until the final prediction. Consequently, the computation is unevenly allocated among images by adjusting the pruned token number, yielding a considerable improvement in computation efficiency. We also observe that pruned easy tokens are primarily located at the central area of objects, while kept hard tokens are located on the boundaries between objects, similar to the segmentation process by humans. Some visualized predictions are juxtaposed in Figure. 5.5.

5.4 Further Discussions

5.4.1 More Visualized Results

The computation is unevenly allocated among different images when applying the proposed DToP, which attributes computation cost to dissimilar recognition difficulties. We present visualized examples for a simple illustration in Figure. 5.6. We see that the reduction of computation cost in GFLOPs can be as high as 57.7% in simple-scene

Method	Backbone	p_0	ADE20K mIoU(%)	ADE20K GFLOPs	Pascal Context mIoU(%)	Pascal Context GFLOPs	COCO-Stuff-10K mIoU(%)	COCO-Stuff-10K GFLOPs
SETR [Zheng et al., 2021b]	ViT-Base	-	47.0	107.7	58.1	92.4	41.2	107.7
+ DToP@Finetune	ViT-Base	0.90	46.3	72.5	57.5	61.4	40.6	77.6
+ DToP@Finetune	ViT-Base	0.98	47.0	80.6	**58.2**	69.1	40.9	86.4
SegViT [Zhang et al., 2022a]	ViT-Large	-	53.3	617.0	63.0	315.4	47.4	366.9
+ DToP@Finetune	ViT-Large	0.90	52.4	380.3	62.2	206.1	46.6	253.1
+ DToP@Finetune	ViT-Large	0.95	52.8	412.8	62.7	224.3	47.1	276.2

TABLE 5.7. **Main results on three semantic segmentation benchmarks.** We apply the proposed DToP with the finetuning training scheme to current state-of-the-art semantic segmentation networks based on plain vision transformers. GFLOPs is the average number of the whole validation dataset. We perform token pruning at $\{8^{th}, 16^{th}\}$ layers for ViT-Large.

FIGURE 5.5. **Visualized results.** The segmentation results are predicted on ADE20K (first row), Pascal Context (middle row), and COCO-Stuff-10K (last row). The model is SegViT with DToP@Finetune based on ViT-Large.

images, such as the example in the first row that contains only the building and sky. For complex-scene images where the object number increases and the scale varies, fewer tokens trigger the early exit, and less GFLOPs reduction is obtained. Even though the computation cost fluctuates among images, the segmentation accuracy remains stable compared with the baseline results.

5.4.2 Symmetrical Downsampling

DToP serves as an unsymmetrical downsampling operator by making an early exit of easy tokens. We compare several commonly used symmetrical downsampling operators, including stride convolution, average pooling, and the nearest sampling. We apply these operators at the end of the 9th layer of ViT-Base [Dosovitskiy et al., 2021a] backbone to ensure an approximate computation overhead. Results with the ADE20K dataset [Zhou et al., 2017b] are shown in Table 5.8. The proposed DToP outperforms others by a large margin.

5.5 Per-Category Results

We present in Figure 5.7 per-category scores on the Pascal Context [Mottaghi et al., 2014a] dataset as an example. We observe that DToP yields negligible performance impact on each category as it *finalizes* easy tokens' predictions instead of *discarding* then *rebuilding* them and making predictions at the final layer.

Methods	GFLOPs	mIoU (%)
Baseline	109.9	49.7
Conv, *stride* = 2	88.4	44.8
2×2 average pool	87.8	44.4
2×2 nearest sampling	87.8	46.1
DToP (ours)	86.8	**49.8**

TABLE 5.8. Comparisons to standard symmetrical downsampling methods under similar computation budget. All methods except baseline follow the @Finetune training scheme.

5.6 Why Plain ViT

We focus our pruning method for semantic segmentation on the plain ViT backbone [Dosovitskiy et al., 2021a] as it offers several advantages over pyramid structures [Liu et al., 2021] or efficient transformers. The plain ViT structure has the potential to unify multiple dense prediction tasks and can be improved with more flexible self-supervised methods [He et al., 2022; Xie et al., 2022]. Additionally, it is capable of connecting visual and language inputs, allowing zero-/few-shot and continuous learning for dense prediction tasks. The proposed dynamic pruning method enables a new paradigm for using ViT in the future. This approach allows for a foundation ViT to be trained on large-scale datasets yet still be applied to various local datasets with flexible computational reduction.

5.7 Conclusion

This work studies the problem of reducing computation costs for existing semantic segmentation based on plain vision transformers. A Dynamic Token Pruning paradigm is proposed based on the early exit of tokens. Motivated by the coarse-to-fine segmentation process by humans, we assume that different tokens representing image regions have dissimilar recognition difficulties and grade all tokens' difficulty levels using the inherent auxiliary blocks. To this end, we finalize the predictions of easy tokens at intermediate layers and halt their forward propagation, which dynamically reduces computation. We further propose a strategy to uphold context information by preserving extremely easy semantic categories after token pruning. Extensive experimental results suggest that the proposed method achieves compelling performance.

Similar to all other dynamic networks, DToP can not take full advantage of the calculation efficiency of a mini-batch. We will make optimization in the future and further expedite vision transformers using the proposed DToP.

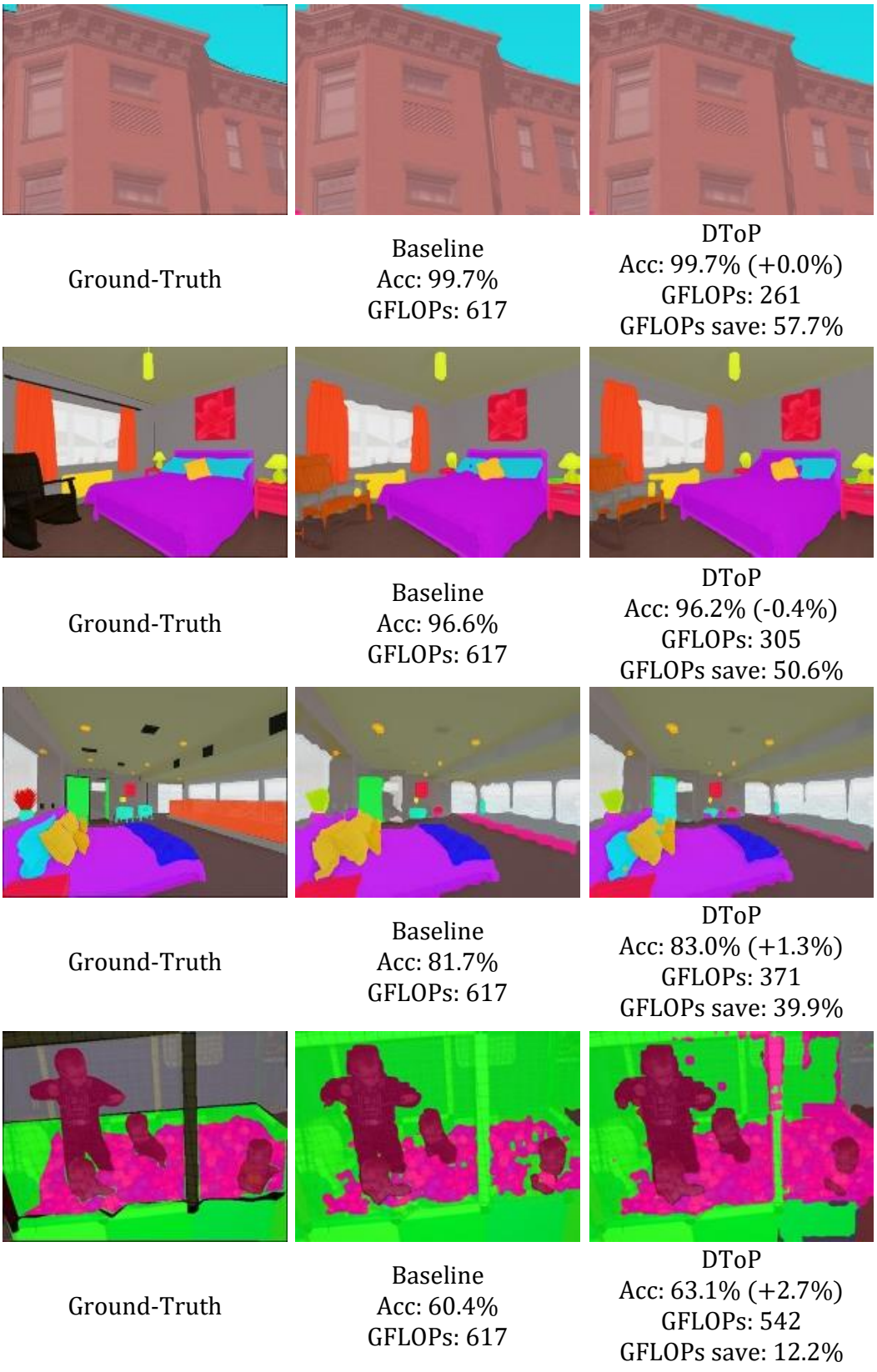


FIGURE 5.6. Visualised examples using ADE20K dataset. As computing the intersection over union (IoU) is unreasonable within a single image, we use the category-agnostic pixel accuracy (Acc) instead. GFLOPs means float-point operations in Giga. Best viewed in color.

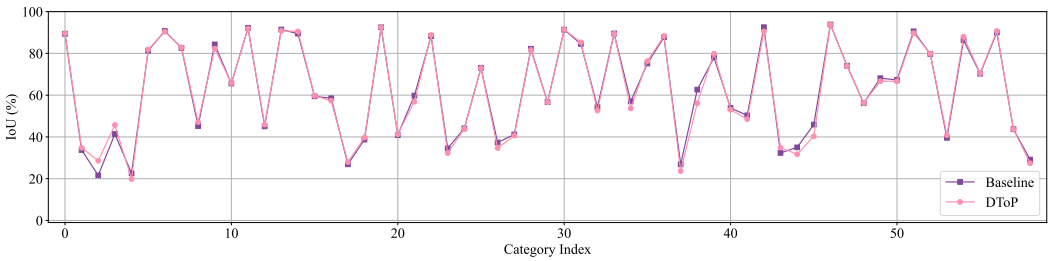


FIGURE 5.7. Per-category scores on Pascal Context dataset with 59 classes excluding *background*.

Chapter 6

Conclusions

In this thesis, we have proposed two methods for semantic segmentation, each designed for different types of semantic segmentation decoders (pyramid and plain). Each method specifically targets and aims to solve the most concerning problems caused by those encoders.

First, We have proposed a compact yet very effective decoder, termed Neural Representational Decoders (NRD), for the semantic segmentation task. For the first time, we use the idea of neural representations for designing the segmentation decoder, which is able to better exploit the structure in the semantic segmentation label space. To implement this idea, we dynamically generate the neural representations with dynamic convolution filter networks so that the neural representations can be incorporated into the standard encoder-decoder segmentation architectures, enabling end-to-end training. We show on a number of semantic segmentation benchmarks that our method is highly efficient and achieves state-of-the-art accuracy.

Next, we present SegViT, a novel approach for semantic segmentation using plain ViT transformer base models. The proposed method introduces a lightweight decoder head that incorporates the Attention-to-mask (ATM) module. Additionally, a *Shrunk* structure is proposed to reduce the computational cost of the ViT encoder by 50% while maintaining competitive segmentation accuracy. Moreover, this work extends the SegViT framework to address the challenge of continual semantic segmentation, aiming to achieve nearly zero forgetting. By protecting the parameters of old tasks, SegViT effectively mitigates the impact of catastrophic forgetting. Extensive experimental evaluations conducted on various benchmarks demonstrate the superiority of SegViT over UPerNet, while significantly reducing computational costs. The introduced decoder head provides a robust and cost-effective solution for future research in the field of ViT-based semantic segmentation.

Finally, we study the problem of reducing computation costs for existing semantic segmentation based on plain vision transformers. A Dynamic Token Pruning paradigm is proposed based on the early exit of tokens. Motivated by the coarse-to-fine segmentation process by humans, we assume that different tokens representing image regions have dissimilar recognition difficulties and grade all tokens' difficulty levels using the inherent auxiliary blocks. To this end, we finalize the predictions of easy tokens at intermediate layers and halt their forward propagation, which dynamically reduces computation. We further propose a strategy to uphold context information by preserving extremely easy semantic categories after token pruning. Extensive experimental results suggest that the proposed method achieves compelling performance. Similar to all other dynamic networks, DToP can not take full advantage of the calculation efficiency of a mini-batch. We will make optimization in the future and further expedite vision transformers using the proposed DToP.

The methods proposed in this thesis have the potential to significantly impact the current state of semantic segmentation, as evidenced by the growing number of works

built upon our contributions. We envision our proposed methods laying a robust foundation for various tasks and applications that rely on semantic understanding, while also offering fresh insights to the research community. Our hope is that these advancements will foster further progress and lead to more impactful developments in the field of semantic segmentation.

Bibliography

Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip Torr. Higher order conditional random fields in deep neural networks. In *Proc. Eur. Conf. Comp. Vis.*, pages 524–540, 2016.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=p-BhZSz59o4.

Walid Bousselham, Guillaume Thibault, Lucas Pagano, Archana Machireddy, Joe Gray, Young Hwan Chang, and Xubo Song. Efficient self-ensemble framework for semantic segmentation. *arXiv preprint arXiv:2111.13280*, 2021.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018a.

Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1209–1218, 2018b.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020a.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. Eur. Conf. Comp. Vis.*, pages 213–229. Springer, 2020b.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Modeling the background for incremental learning in semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9230–9239, 2020.

Sungmin Cha, YoungJoon Yoo, Taesup Moon, et al. Ssul: Semantic segmentation with unknown label for exemplar-based class-incremental learning. In *Proc. Advances in Neural Inf. Process. Syst.*, volume 34, pages 10919–10930, 2021.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):834–848, 2017a.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv: Comp. Res. Repository*, abs/1706.05587, 2017b.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017c.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. Eur. Conf. Comp. Vis.*, 2018a.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 801–818, 2018b.

Wanli Chen, Xinge Zhu, Ruoqi Sun, Junjun He, Ruiyu Li, Xiaoyong Shen, and Bei Yu. Tensor low-rank reconstruction for semantic segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 52–69. Springer, 2020.

Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context autoencoder for self-supervised representation learning. *arXiv preprint arXiv:2202.03026*, 2022.

Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2061–2070, 2023.

Zhiyuan Chen and B. Liu. *Lifelong Machine Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning, 2016.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021a.

Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Proc. Advances in Neural Inf. Process. Syst.*, 34, 2021b.

Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021c.

Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022a.

Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022b.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Proc. Advances in Neural Inf. Process. Syst.*, 34, 2021.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3213–3223, 2016.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. *arXiv preprint arXiv:2302.05442*, 2023.

Henghui Ding, Xudong Jiang, Bing Shuai, Ai Qun Liu, and Gang Wang. Context contrasted feature and gated multi-scale aggregation for scene segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2393–2402, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021a.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *Proc. Int. Conf. Learn. Representations*, 2021b.

Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Proc. Eur. Conf. Comp. Vis.*, pages 86–102. Springer, 2020.

Arthur Douillard, Yifu Chen, Arnaud Dapogny, and Matthieu Cord. Plop: Learning without forgetting for continual semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2021.

Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9285–9295, 2022.

Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3146–3154, 2019a.

Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3146–3154, 2019b.

Junjun He, Zhongying Deng, and Yu Qiao. Dynamic multi-scale filters for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 3562–3572, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 770–778, 2016b.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *Proceedings of International Conference on Learning Representations*, 2017.

Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 603–612, 2019.

Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc van Gool. Dynamic filter networks. In *Proc. Advances in Neural Inf. Process. Syst.*, 2016.

Zhenchao Jin, Tao Gong, Dongdong Yu, Qi Chu, Jian Wang, Changhu Wang, and Jie Shao. Mining contextual information beyond image for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 7231–7241, 2021a.

Zhenchao Jin, Bin Liu, Qi Chu, and Nenghai Yu. Isnet: Integrate image-level and semantic-level context for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 7189–7198, 2021b.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 16071–16080, 2022.

Tsung-Wei Ke, Jyh-Jing Hwang, Ziwei Liu, and Stella Yu. Adaptive affinity fields for semantic segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 587–602, 2018.

Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollar. Panoptic feature pyramid networks. 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *Proc. Eur. Conf. Comp. Vis.*, pages 620–640. Springer, 2022.

Xia Li, Zhisheng Zhong, Jianlong Wu, Yibo Yang, Zhouchen Lin, and Hong Liu. Expectation-maximization attention networks for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 9167–9176, 2019.

Xia Li, Yibo Yang, Qijie Zhao, Tiancheng Shen, Zhouchen Lin, and Hong Liu. Spatial pyramid based graph reasoning for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 8950–8959, 2020a.

Xiangtai Li, Ansheng You, Zhen Zhu, Houlong Zhao, Maoke Yang, Kuiyuan Yang, Shaohua Tan, and Yunhai Tong. Semantic flow for fast and accurate scene parsing. In *Proc. Eur. Conf. Comp. Vis.*, pages 775–793, 2020b.

Xiangtai Li, Houlong Zhao, Lei Han, Yunhai Tong, Shaohua Tan, and Kuiyuan Yang. Gated fully fusion for semantic segmentation. In *Proc. AAAI Conf. Artificial Intell.*, volume 34, pages 11418–11425, 2020c.

Xiaoxiao Li, Ziwei Liu, Ping Luo, Chen Change Loy, and Xiaoou Tang. Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3193–3202, 2017.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40:2935–2947, 2018.

Weicong Liang, Yuhui Yuan, Henghui Ding, Xiao Luo, Weihong Lin, Ding Jia, Zheng Zhang, Chao Zhang, and Han Hu. Expediting large-scale vision transformer for dense prediction without fine-tuning. *arXiv preprint arXiv:2210.01035*, 2022a.

Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *Proc. Int. Conf. Learn. Representations*, 2022b.

Fangjian Lin, Zhanhao Liang, Junjun He, Miao Zheng, Shengwei Tian, and Kai Chen. Structtoken: Rethinking semantic segmentation with structural prior. *arXiv: Comp. Res. Repository*, 2022.

Guosheng Lin, Chunhua Shen, Anton van den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3194–3203, 2016.

Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1925–1934, 2017a.

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 2117–2125, 2017b.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 2980–2988, 2017c.

Jianbo Liu, Junjun He, Jiawei Zhang, Jimmy Ren, and Hongsheng Li. EfficientFCN: Holistically-guided decoding for semantic segmentation. In *Proc. Eur. Conf. Comp. Vis.*, 2020.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 10012–10022, 2021.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015a.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3431–3440, 2015b.

Chenyang Lu, Daan de Geus, and Gijs Dubbelman. Content-aware token sharing for efficient semantic segmentation with vision transformers. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 23631–23640, 2023.

Haoyu Lu, Nanyi Fei, Yuqi Huo, Yizhao Gao, Zhiwu Lu, and Ji-Rong Wen. Cots: Collaborative two-stream vision-language pre-training model for cross-modal retrieval. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 15692–15701, 2022.

Andrea Maracani, Umberto Michieli, Marco Toldo, and Pietro Zanuttigh. Recall: Replay-based continual learning in semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2021.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 4460–4470, 2019.

Umberto Michieli and Pietro Zanuttigh. Incremental learning techniques for semantic segmentation. pages 3205–3212, 2019.

Umberto Michieli and Pietro Zanuttigh. Continual semantic segmentation via repulsion-attraction of sparse and disentangled latent representations. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1114–1124, 2021.

Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. IEEE, 2016.

Tom Mitchell. *Machine learning*. McGraw-hill New York, 1997.

MMSegmentation. MMSegmentation: OpenMMLab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 891–898, 2014a.

Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 891–898, 2014b.

Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 1520–1528, 2015.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Oleksiy Ostapenko, Timothee Lesort, Pau Rodríguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Continual learning with foundation models: An empirical study of latent replay. In *Conference on Lifelong Learning Agents*, pages 60–91. PMLR, 2022.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. *arXiv: Comp. Res. Repository*, 2, 2020.

Yuxin Peng, Jinwei Qi, Zhaoda Ye, and Yunkan Zhuo. Hierarchical visual-textual knowledge distillation for life-long correlation learning. *Int. J. Comput. Vision*, 129:921–941, 2021.

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. BEiT v2: Masked image modeling with vector-quantized visual tokenizers. 2022.

Minh Hieu Phan, Son Lam Phung, Long Tran-Thanh, Abdesselam Bouzerdoum, et al. Class similarity weighted knowledge distillation for continual semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 16866–16875, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *Proc. Int. Conf. Learn. Representations*, 2022.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 12179–12188, 2021.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. In *Proc. Advances in Neural Inf. Process. Syst.*, volume 34, pages 13937–13949, 2021.

Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Proc. Advances in Neural Inf. Process. Syst.*, 34:12786–12797, 2021.

Chenze Shao and Yang Feng. Overcoming catastrophic forgetting beyond continual learning: Balanced training for neural machine translation. *arXiv preprint arXiv:2203.03910*, 2022.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Proc. Advances in Neural Inf. Process. Syst.*, 33, 2020.

Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv: Comp. Res. Repository*, 2021.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 7262–7272, 2021.

Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv: Comp. Res. Repository*, 2019.

Zhi Tian, Tong He, Chunhua Shen, and Youliang Yan. Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3126–3135, 2019.

Zhi Tian, Bowen Zhang, Hao Chen, and Chunhua Shen. Instance and panoptic segmentation using conditional convolutions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022.

Hugo Touvron, Matthieu Cord, and Hervé Jégou. Deit iii: Revenge of the vit. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 516–533. Springer, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Proc. Advances in Neural Inf. Process. Syst.*, 30, 2017a.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017b.

Jiaqi Wang, Kai Chen, Rui Xu, Ziwei Liu, Chen Change Loy, and Dahua Lin. CARAFE: Content-aware reassembly of features. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 3007–3016, 2019.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10):3349–3364, 2020.

Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *Proc. Winter Conf. on Appl. of Comp. Vis.*, pages 1451–1460, 2018.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 568–578, 2021a.

Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, 34:11960–11973, 2021b.

Zhen Wang, Liu Liu, Yiqun Duan, Yajing Kong, and Dacheng Tao. Continual learning with lifelong vision transformer. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 171–181, 2022a.

Zhen Wang, Liu Liu, Yajing Kong, Jiaxian Guo, and Dacheng Tao. Online continual learning with contrastive vision transformer. In *Proc. Eur. Conf. Comp. Vis.*, pages 631–650. Springer, 2022b.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 631–648. Springer, 2022c.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022d.

Longhui Wei, Lingxi Xie, Wengang Zhou, Houqiang Li, and Qi Tian. Mvp: Multimodality-guided visual pre-training. In *Proc. Eur. Conf. Comp. Vis.*, pages 337–353. Springer, 2022.

Tianyi Wu, Yu Lu, Yu Zhu, Chuang Zhang, Ming Wu, Zhanyu Ma, and Guodong Guo. Ginet: Graph interaction network for scene parsing. In *Proc. Eur. Conf. Comp. Vis.*, pages 34–51. Springer, 2020.

Tongtong Wu, Massimo Caccia, Zhuang Li, Yuan-Fang Li, Guilin Qi, and Gholamreza Haffari. Pretrained language model in continual learning: A comparative study. In *Proc. Int. Conf. Learn. Representations*, 2022a.

Yu-Huan Wu, Yun Liu, Xin Zhan, and Ming-Ming Cheng. P2t: Pyramid pooling transformer for scene understanding. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022b.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proc. Eur. Conf. Comp. Vis.*, pages 418–434, 2018.

Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Proc. Advances in Neural Inf. Process. Syst.*, 34, 2021.

Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 3014–3023, 2021.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10809–10818, 2022.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 1857–1866, 2018a.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 325–341, 2018b.

Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018c.

Changqian Yu, Yifan Liu, Changxin Gao, Chunhua Shen, and Nong Sang. Representative graph neural network. In *Proc. Eur. Conf. Comp. Vis.*, pages 379–396, 2020a.

Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 12416–12425, 2020b.

Changqian Yu, Jingbo Wang, Changxin Gao, Gang Yu, Chunhua Shen, and Nong Sang. Context prior for scene segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12416–12425, 2020c.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.

Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In *Proc. Eur. Conf. Comp. Vis.*, pages 173–190. Springer, 2020.

Bowen Zhang, Zhi Tian, Chunhua Shen, et al. Dynamic neural representational decoders for high-resolution semantic segmentation. volume 34, 2021a.

Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan Liu. Segvit: Semantic segmentation with plain vision transformers. In *Proc. Advances in Neural Inf. Process. Syst.*, 2022a.

Chang-Bin Zhang, Jia-Wen Xiao, Xialei Liu, Ying-Cong Chen, and Ming-Ming Cheng. Representation compensation networks for continual semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7053–7064, 2022b.

Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 7151–7160, 2018.

Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. Co-occurrent features in semantic segmentation. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 548–557, 2019.

Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *Proc. Advances in Neural Inf. Process. Syst.*, 34, 2021b.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017a.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017b.

Hengshuang Zhao, Yi Zhang, Shu Liu, Jianping Shi, Chen Change Loy, Dahua Lin, and Jiaya Jia. PSANet: Point-wise spatial attention network for scene parsing. In *Proc. Eur. Conf. Comp. Vis.*, pages 267–283, 2018.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 6881–6890, 2021a.

Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021b.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 633–641, 2017a.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017b.

Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *Proc. Int. Conf. Learn. Representations*, 2022.

Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. Unet++: A nested U-net architecture for medical image segmentation. In *Proc. Deep Learning in Medical Image Analysis Workshop*, pages 3–11, 2018.

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=gZ9hCDWe6ke.

Zhen Zhu, Mengde Xu, Song Bai, Tengteng Huang, and Xiang Bai. Asymmetric non-local neural networks for semantic segmentation. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 593–602, 2019.